Knowledge and Pre-trained Language Models Inside and Out: a deep-dive into datasets and external knowledge

Chenyang Lyu, B.Eng.

A Dissertation submitted in fulfilment of the

requirements for the award of

Doctor of Philosophy

to the

Dublin City University

School of Computing

Supervisors

Prof. Jennifer Foster, Dublin City University

Prof. Yvette Graham, Trinity College Dublin

August 2023

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Sign:  Student No.: 19213991 Date: 30 August 2023

(Chenyang Lyu)

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Dr. Jennifer Foster and Dr. Yvette Graham, for their guidance and support throughout my PhD journey. Their insightful feedback and encouragement have been invaluable in shaping my research in Natural Language Processing.

Moreover, I am grateful for the opportunities and resources provided by ML-Labs and Dublin City University, where I have been able to freely explore my research interests in topics of Natural Language Processing especially Pre-trained Large Language Models. The research community at both ML-Labs and Dublin City University has been truly inspiring, and I've been able to collaborate with some amazing researchers with their unique perspectives on Natural Language Processing including Prof. Cathal Gurrin and Dr. Liting Zhou. These collaborations have helped me broaden my understanding of the field and its many applications.

Furthermore, I've been lucky enough to have a group of awesome PhD students at ML-Labs and Dublin City University to support me along the way. They have been supporting me during the tough times. I would like to express my thanks to Xuehao Liu, Na Li, Qin Ruan, Liang Xu, Pintu Lohar, Rudali Huidrom, Kanishk Verma, James Barry.

Additionally, I would like to give a special thanks to Dr. Tianbo Ji and Dr. Longyue Wang, who have been working with me on many exciting projects and help me gain knowledge of conducting research.

Finally, I would like to express to my parents and friends for their generous support and encouragement, which has helped me navigate the ups and downs of PhD life.

Contents

List of Figures

List of Tables

List of Abbreviations

NLP	Natural Language Processing
PLMs	Pre-trained Language Models
LLMs	Large Language Models
QA	Question Answering
QG	Question Generation
SA	Sentiment Analysis
CBOW	Continuous Bag of Words
SRL	Semantic Role Labeling
VQA	Visual Question Answering
VideoQA	Video Question Answering
EVQA	Event-level Video Question Answering
NNLM	Neural Network Language Model
LSTM	Long Short-Term Memory
BPE	Byte Pair Encoding
DP	Dependency Parsing
NER	Named Entity Recognition
EM	Exact Match

Knowledge and Pre-trained Language Models Inside and Out: a deep-dive into datasets and external knowledge

Chenyang Lyu

Abstract

Pre-trained Language Models (PLMs) have greatly advanced the performance of various NLP tasks and have undoubtedly been serving as the foundation of this field. These pre-trained models are able to capture rich semantic patterns from large-scale text corpora and learn high-quality representations of texts. However, such models still have shortcomings - they underperform when faced with tasks that requires implicit external knowledge to be understood, which is difficult to learn with commonly employed pre-training objectives. Moreover, there lacks a comprehensive understanding of PLMs' behavior in learning knowledge during the fine-tuning phase. Therefore, in order to address the aforementioned challenges, we propose a set of approaches to inject external knowledge into PLMs and demonstrate experiments investigating their behavior of learning knowledge during the fine-tuning phase, primarily focusing on Sentiment Analysis, Question Answering and Video Question Answering.

Specifically, we introduce novel approaches explicitly using textual historical reviews of users and products for improving sentiment analysis. To overcome the problem of context-question lexical overlap and data scarcity for question generation, we propose a novel method making use of linguistic and semantic knowledge with heuristics. Additionally, we explore how to utilise multimodal (visual and acoustic) information/knowledge to improve Video Question Answering.

Experiments conducted on benchmark datasets show that our proposed approaches achieve superior performance compared to state-of-the-art models, demonstrating the effectiveness of our methods for injecting external knowledge. Furthermore, we conduct a set of experiments investigating the learning of knowledge for PLMs for question answering under various scenarios. Results reveal that the internal characteristics of QA datasets can pose strong bias for PLMs when learning from downstream tasks datasets. Finally, we present an in-depth discussion of future directions for improving PLMs with external knowledge.

Chapter 1

Introduction

Recent years have witnessed the emergence of Pre-trained Language Models (PLMs), such as ELMo, GPT, BERT, XLNet, GPT-3 and InstructGPT [Wang et al., 2018, Peters et al., 2018, Radford et al., 2018, Devlin et al., 2019b, Yang et al., 2019, Brown et al., 2020, Chen et al., 2021, Ouyang et al., 2022, OpenAI, 2303], which have been widely used in many NLP tasks and have shown superior performance compared to previous approaches [Devlin et al., 2019b, Qiu et al., 2020, Brown et al., 2020]. PLMs are firstly pre-trained on large-scale unlabeled text corpora using self-supervised objectives, followed either by 1) fine-tuning on downstream tasks with labeled data using supervised learning or 2) direct prompting to perform a downstream task with examples (few-shot) or without (zero-shot), resulting in new paradigms for NLP research. This has been shown to surpass previous neural approaches trained only on labeled downstream task data [Devlin et al., 2019a]. Diverging from early approaches producing static word embeddings where each word only has one embedding vector, PLMs produce *contextualized word representations* [Peters et al., 2018], where words have different representation vectors within different contexts. This is in line with the commonsense assumption that the semantics of a word should not only depend on itself but also depend on its context. Such modifications, powered with large neural models [Vaswani et al., 2017] and large-scale corpora, give significant improvements on a wide range of NLP tasks including sentiment analysis, question answering and natural language inference. Probing tasks have shown that the representations learned by PLMs capture aspects of the semantics and syntax of language [Jawahar et al., 2019, Rogers et al., 2020]. Furthermore, recent advancements in PLMs such as

GPT-3 and InstructGPT [Brown et al., 2020, Ouyang et al., 2022, Bang et al., 2023] have again significantly improved the performance of various tasks. Despite the huge success of pre-trained models in NLP, these models can still lack the knowledge needed for tasks which require information beyond the text, such as sentiment analysis, entity typing and question answering [Da and Kasai, 2019, Liu et al., 2020a]. The incorporation of structured knowledge from knowledge graphs has been explored in [Zhang et al., 2019, Yu et al., 2020, He et al., 2020, Colon-Hernandez et al., 2021, Wang et al., 2021c], yielding improvements for various knowledge-intensive tasks including named-entity recognition, relation classification, entity typing and question answering especially for domains such as medicine. For example, in entity typing and relation classification, without external entity knowledge such as knowledge base triples of the form *<Entity1, Relation, Entity2>*, it is difficult for a pre-trained language model to produce the correct prediction even though it has captured rich information from pre-training on huge volumes of unstructured text. In [Zhang et al., 2019], the use of entity information from knowledge graphs injected into the joint pre-training process in [Devlin et al., 2019a] substantially improves model performance on entity-typing and relation extraction, where the token-entity alignment objective aims to inject the entity information into the representations learned by the transformer encoder. While more powerful PLMs such as GPT-4 [OpenAI, 2303] exhibit significantly better performance compared to previous PLMs, they still underperform on certain tasks such as multi-step reasoning, numerical reasoning, tasks needing common sense knowledge, as well as low-resource languages [Bang et al., 2023, Lai et al., 2023].

Earlier work mainly focuses on incorporating structured knowledge from knowledge graphs (entity knowledge and linguistic knowledge). An exploration of methods for injecting other external information, beyond that found in text, is lacking. For example, knowledge of personalized preference is useful for sentiment analysis as sentiment conveyed by texts can be highly personalized. Approaches for incorporating knowledge have been limited to learning joint representations of text and knowledge, requiring substantial modifications to model architecture. Therefore,

we focus on exploring novel approaches incorporating knowledge beyond text such as semantic knowledge, personalized preferences and multi-modal information into PLMs while minimizing model architecture modifications. Furthermore, despite the success as well as the large volume of research conducted on PLMs [Qiu et al., 2020, Zhang et al., 2020b], less emphasis has been placed on the effects of the data used for fine-tuning. A better understanding of the data has the potential to improve the generalizability of models [Rogers, 2021, Gardner et al., 2021], as well as providing helpful information for constructing datasets [Bender and Friedman, 2018, Geva et al., 2019]. Thus, we will explore three major research questions, 1) how can we utilize the extra information in the metadata of product reviews to improve document-level sentiment analysis? 2) how can we leverage linguistic and semantic knowledge to improve Unsupervised Question Answering, and understand the role of QA data in neural model learning? 3) how can the utilization of multi-modal information improve Video Question Answering tasks for Pre-trained Vision-Language Models?

Through these research questions, we aim to advance the understanding and practical application of incorporating knowledge beyond text into PLMs for three specific tasks including Sentiment Analysis, Question Answering and Video Question Answering. Additionally, we recognize the importance of understanding the effects of fine-tuning data and believe that our findings can contribute to improving model generalizability and providing insights for dataset construction.

1.1 Research Questions

The primary goal of this thesis is to investigate how to use **external knowledge**, beyond the normal fine-tuning data that is commonly employed, to improve the performance of PLMs on downstream tasks that may require implicit external knowledge. More specifically, we focus on three tasks: Document-level Sentiment Analysis, Question Answering and Video Question Answering. Therefore, we propose

three research questions

RQ1: How can we utilize the extra information in the metadata of product reviews to improve document-level sentiment analysis?

The goal of Sentiment Analysis is to predict the sentiment conveyed by a piece of opinionated text (often a review). In document-level sentiment analysis with user and product information, we also know the user who wrote the review and the product being evaluated by the review. User and product context can be helpful for predicting the correct sentiment label: the same user may tend to use the same or a highly similar narrative style as well as similar word choices when writing reviews. For example, a user who has high expectations for the product being evaluated might use words like *good, nice* but only give a rating *medium positive* or even use such positive words sarcastically to give a negative rating; similarly, the reviews belonging to a particular product may have the same group of opinionated words and narrative style towards the product being evaluated. Earlier work [Tang et al., 2015b, Chen et al., 2016b, Ma et al., 2017, Dou, 2017, Long et al., 2018, Amplayo, 2019, Amplayo et al., 2018] mainly focuses on modeling users and products as embedding vectors which are updated in the training process, with the expectation that such embedding vectors can implicitly learn the bias introduced by users and products. However, such approaches fail to fully make use of the textual information of historical reviews belonging to a user or a product, since it is difficult to learn meaningful representations of users and products if they are only updated and learned by back propagation, especially for users and products who only have small number of reviews. Therefore, RQ1 will focus on how to model the historical reviews of a user and product to learn more meaningful representations of user and product context for the purpose of improving the prediction of sentiment labels.

RQ2: How can we leverage linguistic and semantic knowledge to improve unsupervised Question Answering, and understand the role of QA data in neural model learning?

The goal of Question Generation (QG) is to generate plausible questions for given <*passage, answer*> pairs. QG can be applied in dialogue systems as well as educational applications [Graesser et al., 2005] and as a data augmentation method for Question Answering (QA) [Puri et al., 2020]. There are two classes of QG approaches: 1) *Template-based QG* [Heilman and Smith, 2009, 2010], which uses heuristics induced from linguistic knowledge to transform declarative sentences into questions; 2) *Supervised QG* [Du et al., 2017, Duan et al., 2017, Zhang and Bansal, 2019, Chen et al., 2019, Xie et al., 2020, Ma et al., 2020, Ji et al., 2021], which uses existing QA datasets to train a QG system.

Moreover, after the emergence of PLMs, substantial improvements have been obtained on many NLP tasks, such as QA [Qiu et al., 2020, Bommasani et al., 2021]. However, we still cannot neglect the importance of the dataset, and indeed, this has become a new focus of NLP research [Søgaard et al., 2021, Lewis et al., 2021, Liu et al., 2021b]. RQ2 is focused on QA and will explore how to combine the advantages of the template-based and supervised QG methods, while addressing their shortcomings, and investigate how the generated questions can be used to train an unsupervised QA system. We will also analyze how a pre-trained model learns from QA datasets.

RQ3: How can the utilization of multi-modal information improve Video Question Answering tasks for Pre-trained Vision-Language Models?

Video Question Answering (VideoQA) [Lei et al., 2018, Xu et al., 2021b] is a challenging task that aims to interpret visual information and answer natural language questions about video content. Despite recent advancements in Pre-trained

Vision-Language Models for multi-modal NLP tasks, such models still face significant challenges in handling complex multi-modal information, such as visual and audio content [Zhong et al., 2022b]. To overcome these challenges and improve the performance of pre-trained language models in VideoQA tasks, it is crucial to investigate the potential of utilizing complex multi-modal information. By integrating visual and audio features into pre-trained vision-language models, it is possible to enhance their ability to understand complex video content and answer associated questions more accurately. However, effectively utilizing multi-modal information remains very challenging, and various approaches have been developed to address this challenge. RQ3 aims to explore how the utilization of multi-modal modality information can improve video question answering tasks for Pre-trained Vision-Language Models.

1.2 Thesis Outline

This thesis is focused on exploring the learning of knowledge for Pre-trained Large Language Models, for Sentiment Analysis and Question Answering. The remainder of the thesis is organized as follows:

- Chapter 2 outlines related work on Pre-trained Large Language Models including various variants of PLMs and knowledge-enhanced PLMs, demonstrating the evolution of PLMs from multiple dimensions. We also include related research on Document-level Sentiment Analysis with user and product context, Question Answering and Multi-modal Question Answering.

- Chapter 3 presents our proposed approaches on incorporating the textual information of historical reviews belonging to the same user and product for improving Document-level Sentiment Analysis. We conduct extensive experiments to validate the effectiveness of our approaches, with a view to answering *RQ1: How can we utilize the extra information in the metadata of product reviews to improve document-level sentiment analysis?*

- Chapter 4 presents a set of experiments aiming to address *RQ2: How can we leverage linguistic and semantic knowledge to improve Unsupervised Question Answering, and understand the role of QA data in neural model learning?* We first describe our approach of utilizing linguistic and semantic knowledge for improving Unsupervised Question Answering via summarization-informed Question Generation. We discuss the details of our approach, particularly how we manipulate the semantic roles in the summary sentence to transform it to an interrogative sentence. The linguistic and semantic knowledge is explicitly incorporated in the dataset for Question Generation, resulting in high-quality synthetic data for Unsupervised Question Answering. Furthermore, to understand how neural QA systems learn from QA dataset during fine-tuning phase we design experiments to investigate the effect of internal characteristics on a QA system's performance.

- Chapter 5 tries to provide answers to *RQ3: How can the utilization of multi-modal information improve Video Question Answering tasks for Pre-trained Vision-Language Models?* We describe our proposed approaches aiming to effectively incorporate multi-modal information for improving Video Question Answering. We present the experiments conducted on a set of benchmark datasets to study how the incorporation of multi-modal information affects Video Question Answering.

- Chapter 6 summarises the thesis content and presents conclusions drawn from the experiments and results included in the thesis. We also discuss the potential promising future directions for research on Pre-trained Large Language Models.

1.3 Publications

The work in this thesis has been published in several papers. The content of Chapter 3 has been published in two papers on incorporating user and product information for Sentiment Analysis:

- **Chenyang Lyu**, Jennifer Foster, Yvette Graham. Improving Document-Level Sentiment Analysis with User and Product Context, In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020 (Oral presentation)*

- **Chenyang Lyu**, Linyi Yang, Yue Zhang, Yvette Graham, and Jennifer Foster. Exploiting Rich Textual User-Product Context for Improving Sentiment Analysis. In *Findings of the 61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*

There is another paper on Sentiment Analysis, but its focus is not on incorporating user and product information, which is not included in the thesis:

- **Chenyang Lyu**, Tianbo Ji, Yvette Graham. Incorporating Context and Knowledge for Better Sentiment Analysis of Narrative Text, In *Proceedings of the Third International Workshop on Narrative Extraction from Texts held in conjunction with the 42nd European Conference on Information Retrieval, ECIR 2020 Workshop*

The major content of Chapter 4 has been published in two papers. One of them is about summarisation-informed question generation for unsupervised question answering, the other one is a paper focusing on analysing and understanding the effect of internal characteristics of QA datasets on model performance:

- **Chenyang Lyu**, Lifeng Shang, Yvette Graham, Jennifer Foster, Xin Jiang and Qun Liu. Improving Unsupervised Question Answering via Summarization-Informed Question Generation, In *Proceedings of The 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021 (Oral presentation)*

- **Chenyang Lyu**, Jennifer Foster, and Yvette Graham. Extending the Scope of Out-of-Domain: Examining QA models in multiple subdomains. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP, ACL 2022*

The research described in Chapter 5 has been published in two papers on Video Question Answering, one is about incorporating semantic knowledge for improving the reasoning of VideoQA systems, the other one focuses on injecting graph-level information from semantic knowledge into VideoQA systems:

- **Chenyang Lyu**, Tianbo Ji, Yvette Graham, and Jennifer Foster. Semantic-aware Dynamic Retrospective-Prospective Reasoning for Event-level Video Question Answering. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2023.*

- **Chenyang Lyu**, Wenxi Li, Tianbo Ji, Longyue Wang, Liting Zhou, Cathal Gurrin, Linyi Yang, Yi Yu, Yvette Graham, and Jennifer Foster. Graph-Based Video-Language Learning with Multi-Grained Audio-Visual Alignment. In *the 31st ACM International Conference on Multimedia, ACM-MM 2023.*

There is another paper relevant to VideoQA; however, since this paper mainly focuses on improving the efficiency of VideoQA systems and is not directly related to incorporating multimodal information, it is not included in the thesis:

- **Chenyang Lyu**, Tianbo Ji, Yvette Graham, and Jennifer Foster. Is a Video worth $n \times n$ Images? A Highly Efficient Approach to Transformer-based Video Question Answering. In *Proceedings of The Third Workshop on Simple and Efficient Natural Language Processing, ACL 2023.*

The paper below initially and significantly inspired my research work on VideoQA. Since this paper doesn't focus on incorporating external knowledge into PLMs, it is not included in this thesis:

- **Chenyang Lyu**, Manh-Duy Nguyen, Van-Tu Ninh, Liting Zhou, Cathal Gurrin, and Jennifer Foster. Dialogue-to-Video Retrieval. In *Proceedings of the 45th European Conference on Information Retrieval, ECIR 2023.*

Chapter 2

Background

In this chapter, we will give an overview of the material which is relevant for this Ph.D. project. In particular, we will first describe the development of pre-training techniques ranging from Word2Vec to ELMo, GPT and BERT to LLMs such as GPT-3 and InstructGPT. Lastly, we will discuss the research which is related to our main research questions concerning how to inject knowledge into Pre-trained Language Models, including specific tasks such as Document-level Sentiment Analysis, Question Answering, Question Generation and Video Question Answering.

2.1 Pre-trained Language Models

There are two dimensions to categorizing pre-training techniques. The first is feature-based pre-training approaches (Word2Vec, GloVe,, ELMo) versus non feature-based approaches (GPT, BERT). The second is non-contextualized word embeddings (Word2Vec, GloVe) versus contextualized word representations (ELMo, GPT, BERT). Some representative approaches will be discussed briefly in the following sections.

2.1.1 Word2vec

Learning meaningful word representations is a long-standing problem [Rumelhart et al., 1986, Hinton et al., 1986, Elman, 1990, Deerwester et al., 1990, Bengio et al., 2003]. Following previous work, Mikolov et al. [2013] proposed Word2Vec which makes use of large text corpora to learn semantically-meaningful word embeddings, Word2Vec significantly improves the quality of word embeddings over a Neural

Network Language Model (NNLM) [Bengio et al., 2003] as demonstrated when measuring syntactic and word semantic similarities. Essentially, there is a word embedding matrix $W \in R^{V \times h}$ in Word2Vec, and each word in vocabulary is projected to a continuous dense vector w with fixed length h. Then there are two training architectures: 1) Continuous Bag of Words (CBOW), where the goal is to predict the correct pivot word based on its surrounding words which is called the *context* 2) Skip-gram model, where the aim is to predict the context words given the pivot word.

In CBOW, the objective is formulated as:

$$P(w_i|w_{i-t}, ..., w_{i-1}, w_{i+1}, ..., w_{i+t}) = \frac{e^{w_c^T w_i}}{\sum_{j=0}^{V} e^{w_c^T w_j}} \tag{2.1}$$

the probability of w_i given the context words $w_{i-t}, ..., w_{i-1}, w_{i+1}, ..., w_{i+t}$ is a probability distribution over all the words in vocabulary, where t is the context window size, w_c is the representation of w_i's context, w_c is computed by:

$$w_c = mean(w_{i-t} + ... + w_{i-1} + w_{i+1} + ... + w_{i+t}) \tag{2.2}$$

i.e w_c is the average of all context word embeddings which can be further fed into a linear layer in practice. The training objective of CBOW is to maximize the probability of predicting the correct word w_i over all samples in the training corpus:

$$J(\theta) = \sum_i log(P(w_i|w_{i-t}, ..., w_{i-1}, w_{i+1}, ..., w_{i+t})) = \sum_i (log(e^{w_c^T w_i}) - log(\sum_{j=0}^{V} e^{w_c^T w_j})) \tag{2.3}$$

In the skip-gram model, the aim is to predict the context words given the pivot word - the inverse of the goal of CBOW. Therefore the probabilistic formulation of the skip-gram model is:

$$P(w_{i-t}, ..., w_{i-1}, w_{i+1}, ..., w_{i+t}|w_i) = \prod_{j=0, j!=t}^{2t} \frac{e^{w_{i-t+j}^T w_i}}{\sum_{k=0}^{V} e^{w_k^T w_i}} \tag{2.4}$$

The training objective of skip-gram is to maximize the probability of predicting the correct context words given a pivot word. Therefore the overall objective of the skip-gram model over all samples in a corpus is:

$$\begin{aligned} J(\theta) &= \sum_{i} log(P(w_{i-t}, ..., w_{i-1}, w_{i+1}, ..., w_{i+t}|w_i)) \\ &= \sum_{i}(\sum_{j=0, j\neq t}^{2t} (log(e^{w_{i-t+j}^T w_i}) - log(\sum_{k=0}^{V} e^{w_k^T w_i}))) \end{aligned} \tag{2.5}$$

In practice, skip-gram is the common choice for the implementation of Word2Vec. The word embeddings of Word2Vec are trained using the skip-gram objective on Google News (approximately 1 billion words)[1]. After pre-training, the word embedding matrix can be used in other tasks as the initial values of word representations. Hence such approaches are called feature-based pretraining methods. Besides Word2Vec, other word embedding models such as GloVe [Pennington et al., 2014], aiming at capturing global word co-occurrence patterns in text corpora, are also widely used in NLP tasks.

2.1.2 Contextualized Word Representations

Although feature-based pretraining techniques such as Word2Vec and GloVe yield high-quality word embeddings which substantially improve the performance of many NLP tasks, there is still a major drawback of such approaches: words tend to have different semantic meanings within different contexts. In other words, the embeddings of a word should be different within different contexts. That suggests the need for context-dependent word representations. However, methods such as Word2Vec only result in context-independent word representations. In order to address such a challenge, contextualized word representations are subsequently proposed. We will introduce three representatives of them: ELMo, GPT and BERT [Peters et al., 2018, Radford et al., 2018, Devlin et al., 2019a] in the following sections.

[1] https://code.google.com/archive/p/Word2Vec/

2.1.2.1 ELMo: Deep Contextualized Word Representations

Peters et al. [2018] propose *ELMo*, a bidirectional neural language model pre-trained on a large text corpus, to address this challenge caused by early word embedding approaches. The forward LM aims to model the probability of a sequence $(w_1, w_2, ..., w_n)$ by computing the probability of each word w_i based on the words appearing before w_i (those words on the left side of w_i in a sentence):

$$P(w_1, w_2, ..., w_n) = \prod_{1}^{n} P(w_i | w_1, ..., w_{i-1}) \tag{2.6}$$

whereas the backward LM computes the probability of a sequence by modeling the probability of word w_i based on the words appearing after w_i (those words on the right side of w_i in a sentence):

$$P(w_1, w_2, ..., w_n) = \prod_{1}^{n} P(w_i | w_{i+1}, ..., w_n) \tag{2.7}$$

A bidirectional LM models the probability of a sequence $(w_1, w_2, ..., w_n)$ by combining the forward LM and the backward LM:

$$P(w_1, w_2, ..., w_n) = \prod_{1}^{n} P(w_i | w_1, ..., w_{i-1}) + \prod_{1}^{n} P(w_i | w_{i+1}, ..., w_n) \tag{2.8}$$

The optimization objective for the parameters θ of ELMo is to maximize the log likelihood of $P(w_1, w_2, ..., w_n)$:

$$J(\theta) = logP(w_1, w_2, ..., w_n) = \sum_{1}^{n} logP(w_i | w_1, ..., w_{i-1}) + \sum_{1}^{n} logP(w_i | w_{i+1}, ..., w_n) \tag{2.9}$$

In the practical implementation of ELMo, a bidirectional LSTM [Hochreiter and Schmidhuber, 1997] is employed as the bidirectional LM, which consumes the context words and produce a representation that can be used to model a distribution over the whole vocabulary step by step. After pre-training on a large text corpus by

jointly optimizing the likelihood of the forward and backward LM. The resulting bidirectional LM can be used to produce contextualized word representations by running it on the text of any specific downstream tasks. Specifically, assuming that the bi-LSTM has L layers, the representations produced can be denoted as $h^F_{i,j}$ and $h^B_{i,j}$, where i represents the i-th word, j is the j-th layer of bi-LSTM and F, B represent the forward LM and backward LM respectively. To make use of these representations, ELMo computes task specific representation h^{ELMo}_i for the i-th word w_i by:

$$h^{ELMo}_i = \lambda \sum_{0}^{L} s_j h_{i,j} \tag{2.10}$$

where λ is a task-specific factor, s_j is the weight coefficient for the representations of layer j and $h_{i,j} = [h^F_{i,j}, h^B_{i,j}]$.

In the pre-training stage, ELMo is trained using the bidirectional LM objective on the 1 Billion Word Benchmark [Chelba et al., 2013]. After ELMo has been pre-trained, as described in [Peters et al., 2018], we can run ELMo on task-specific datasets to obtain contextualized word representations which then can be used in the initial layer (the embedding layer) of any task-specific models: $X'_i = [X_i, h^{ELMo}_i]$, where X_i is the embeddings of the i-th word. By injecting ELMo representations in the embedding layer, Peters et al. [2018] show significant improvements on various NLP tasks including named entity recognition, coreference resolution, semantic role labeling, sentiment analysis, etc. Moreover, it is observed in Peters et al. [2018] that using ELMo in the output layer (same as how ELMo is used in the embedding layer) results in further improvements for some tasks such as SQuAD [Rajpurkar et al., 2016]. ELMo is a contextualized word representation model. However, it is worth noting that when using ELMo in downstream tasks the weights of the bidirectional LM will not be updated and only the word representations produced by the BiLSTM are injected into the task specific model - this is similar to featured-based approaches [Mikolov et al., 2013, Pennington et al., 2014] that provide a word

embedding matrix which is used as the initial word representations. The difference between ELMo and static word embeddings is that the word representations produced by ELMo are context-dependent which has been shown to be helpful in boosting performance of many NLP tasks.

2.1.2.2 GPT: Generative pre-training

Radford et al. [2018] propose a pre-training approach using a neural architecture different from the BiLSTM [Hochreiter and Schmidhuber, 1997] used in ELMo [Peters et al., 2018] called a *transformer* [Vaswani et al., 2017] in which the major component is the multi-head self-attention mechanism, which has been shown to be more effective in modeling long-range dependencies in text compared to recurrent neural networks [Liu et al., 2018]. The GPT model comprises 12 *transformer* blocks, each *transformer* block produces word representations and passes it to next *transformer* block:

$$h_j = transformer\_block(h_{j-1}) \tag{2.11}$$

where j represents the j-th block. Different from the bidirectional LM design in ELMo, GPT uses the causal language model objective, which means GPT only adopts the forward LM. Therefore the objective of GPT becomes the maximization of the following log likelihood for a sequence of words $(w_1, w_2, ..., w_n)$:

$$J(\theta) = logP(w_1, w_2, ..., w_n) = \sum_{1}^{n} logP(w_i|w_1, ..., w_{i-1}) \tag{2.12}$$

In experiments GPT uses BookCorpus [Zhu et al., 2015] as the pre-training text corpus. Radford et al. [2018] propose a new paradigm for NLP tasks: pre-training + finetuning. Firstly we pre-train a large transformer-based language model on text using the self-supervised objective 2.12, then finetune this pre-trained model on downstream supervised tasks. In the finetuning stage, the objective becomes to maximize the probability of label y for the given sequence of words $(w_1, w_2, ..., w_n)$,

which is modeled by adding an extra linear layer to transform the representations produced by *transformer* to logits representing probability distribution over labels:

$$P(y|w_1, w_2, ..., w_n) = linear\_layer(h_L) \tag{2.13}$$

where h_L is the representations in the last *transformer* block, $linear\_layer$ is the extra linear layer added on top of the GPT model, in which the parameters $\tilde{\theta}$ (the pre-trained model parameters θ and the extra parameters θ_{task}) also need to be optimized: $J(\tilde{\theta}) = \sum_j logP(y_j|w_1, w_2, ..., w_n)$. The finetuning stage requires minor modifications to model architecture, because only a few more components need to be added into GPT model [Radford et al., 2018]. As shown in Radford et al. [2018], the GPT model achieves state-of-the-art performance over many NLP tasks including text classification, textual entailment, textual similarity, reading comprehension, especially compared to ELMo. Moreover due to its unidirectional LM objective, it can be used in text generation tasks.

Although the usage of GPT shares some similarities with ELMo, they have significant differences: (i). ELMo uses a bidirectional LM objective whereas GPT adopts a unidirectional LM. (ii) ELMo uses LSTM as its neural architecture and GPT uses *transformer* (iii). In ELMo, the weights of the pre-trained LM are frozen in downstream tasks whereas, in GPT, the pre-trained LM will be finetuned in downstream tasks. (iiii). In the finetuning stage, ELMo provides *word representations* which are *injected* into downstream models without modifications to the architecture of the task-specific model, whereas GPT provides a *model* which only needs a few extra layers added on top of it for downstream tasks.

2.1.2.3 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Following the significant success of PLMs such as ELMo [Peters et al., 2018] and GPT [Radford et al., 2018], Devlin et al. [2019a] propose a new pre-trained model,

BERT, which adopts new objectives in the pre-training stage and has been widely used in NLP research especially in natural language understanding tasks. BERT uses the *transformer* [Vaswani et al., 2017] as its building block, which is the same as the neural architecture used in GPT. To pre-train the BERT model on large text corpora [Devlin et al., 2019a], two objectives - Masked Token Prediction and Next Sentence Prediction are used:

Masked Token Prediction Devlin et al. [2019a] make use of large-scale text corpora to create a masked token prediction objective by masking a certain proportion of tokens in the original sequence and then training BERT model to recover the masked tokens based on the unmasked tokens. Specifically, supposing that for a given sequence $(w_1, w_2, ..., w_n)$, we randomly mask some tokens w in the original sequence by replacing a masked token w with a special token *[MASK]*, the indices of the masked tokens are denoted as $\hat{I}$ and the original indices of all tokens including masked tokens and unmasked tokens are denoted as I, the indices for unmasked tokens are represented as $I - \hat{I}$. We input the edited sequence $(w_1, w_2, ..., w_n)$ in which some tokens are replaced with *[MASK]* to the BERT model and aim to predict the original replaced tokens. Therefore the objective of Masked Token Prediction can be formulated as:

$$J_1(\theta) = logP(\hat{W}|\tilde{W}) = \sum_{i \in \hat{I}} logP(w_i|w_{j_1}, w_{j_2}, ..., w_{j_n}; j_k \in \{I - \hat{I}\}) \tag{2.14}$$

where $\hat{W}$ and $\tilde{W}$ represent masked tokens and unmasked tokens respectively. Note that in BERT the prediction of masked tokens depends on the context on both directions, which differs from the causal language modeling in GPT where the prediction of the next token only depends on the historical context. This is also different from the language modeling objectives in ELMo. Although ELMo adopts a bidirectional LM, it only makes use of the context from a certain direction (either forward or backward) when predicting a word. The design of Masked Token Prediction allows BERT to model language dependencies bidirectionally by utilizing

the information from bidirectional contexts.

Next Sentence Prediction In order to model the dependencies between units larger than words, Devlin et al. [2019a] propose Next Sentence Prediction working with Masked Token Prediction, which concatenates two sentences (A, B), inputs the sequence to BERT model, then predicts whether sentence B is the sentence following sentence A in the original article. The *positive* examples $\{A,B\}$ can be taken from articles in the corpus, *negative* examples $\{A, \tilde{B}\}$ can be created by fixing sentence A and randomly drawing sentence $\tilde{B}$ from the corpus. The optimization objective of Next Sentence Prediction can be formulated as:

$$J_2(\theta) = \sum_{\{A,B\}\in D} logP(y|A,B) + \sum_{\{A,\tilde{B}\}\in \tilde{D}} logP(1-y|A,\tilde{B}) \tag{2.15}$$

where D and $\tilde{D}$ represent the collections of *positive* and *negative* examples respectively, $y \in \{0,1\}$ is the label for whether B is the next sentence of A. If y is the label for *positive* examples, the label for *negative* examples is $1-y$.

The overall objective for the optimization of parameters θ of the BERT model is $J(\theta) = J_1(\theta)+J_2(\theta)$. In experiments, BERT is firstly pre-trained on BookCorpus [Zhu et al., 2015] and English Wikipedia which contain 800 million and 2500 million words respectively. BERT is then transferred to downstream tasks with minor modifications to the model architecture - only a few layers need to be added, according to the experimental results in [Devlin et al., 2019a]. BERT greatly improves the performance on many NLP tasks compared to state-of-the-art approaches, especially on the GLUE benchmark [Wang et al., 2018] where the improvement is 7.7% absolute points. When employing BERT in downstream tasks, the whole model architecture including the word embedding matrix in the lower layer will be used. That is different from Word2Vec/GloVe which only transfers the learned static and context-independent word embeddings to downstream tasks [Mikolov et al., 2013, Pennington et al., 2014] whereas ELMo generates contextulized word representations. Furthermore, what is different between BERT and GPT is: 1) BERT employs Masked Language Modeling

for its pre-training and BERT can utilize bidirectional context to predict the maksed words. 2) whereas GPT uses Causal Language Modeling that auto-regressively predicts the next word only from left to right which means GPT can only uses unidirectional context.

2.1.2.4 BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

Different from the two previous successful PLMs, GPT (a decoder-only model that focuses on generating text auto-regressively) and BERT (an encoder-only model designed for bidirectional context understanding), Lewis et al. [2020] proposed BART – an encoder-decoder model pre-trained under a sequence-to-sequence framework, which combines the strengths of both encoding for context representation and decoding for text generation, thereby enhancing its capabilities across a wide range of NLP tasks. Similar to GPT and BERT, BART is also built on blocks of *transformer* Vaswani et al. [2017] for its encoder and decoder. The pre-training objective of BART is inherited from Auto-Encoders [Hinton and Zemel, 1993, Hinton et al., 2011], which first encodes the input sentence into hidden states and then generates/recovers the original sentence based on the hidden representations. Moreover, to prevent the model from simply learning to copy the original input sentence, Lewis et al. [2020] introduced several perturbation methods to inject noise into the input sentences, including 1) *Token masking*: same as BERT [Devlin et al., 2019a], a certain proportion of tokens are randomly replaced with *[MASK]* tokens. 2) *Token deletion*: a certain proportion of tokens are randomly removed without inserting any special tokens. 3) *Text infilling*: a set of text spans in the input sentence are replaced with a single *[MASK]* token. 4) *Sentence permutation*: the input is divided by full stops into sentences, which are then randomly shuffled. 5) *Document rotation*: the input is rotated so that it begins with a randomly chosen token. With the above perturbation methods, the BART model is trained to generate the original input text, allowing the model to learn word dependencies through a generative sequence-to-sequence

objective. The pre-training objective of BART can be formulated as:

$$J(\theta) = logP(W|\hat{W}) = \sum_{i} logP(w_i|w_0, ..., w_{i-1}, \hat{W}) \tag{2.16}$$

where $\hat{W}$ is the corrupted input text, and W represents the original input text; the overall objective is to generate the original input text word by word, conditioning on the corrupted input and the words generated in the previous steps. We can see that the pre-training of BART differs significantly from BERT. Although the generative objective is similar to GPT, the generation process in GPT only conditions on the previously generated words, whereas BART also conditions on the corrupted input. After being pre-trained on a large collection of books and Wikipedia texts [Liu et al., 2019b] (approximately 160 GB texts), BART shows superior performance on various NLP tasks, including text classification, natural language inference, and machine reading comprehension [Rajpurkar et al., 2016, Vinodhini and Chandrasekaran, 2012], among others. Moreover, BART is capable of text generation due to its sequence-to-sequence architecture, while PLMs such as BERT can hardly be employed for text generation tasks. BART also exhibits strong performance on generation tasks such as summarization and machine translation [Chen et al., 2016a, Narayan et al., 2018, Wu et al., 2016], among others. Furthermore, a multilingual version of BART, mBART, is proposed by Liu et al. [2020b], which is pre-trained on multilingual corpora, demonstrating superior performance on cross-lingual tasks.

2.2 Large Language Models

Recently, Large Language Models (LLMs) with an increased number of parameters and extensive pre-training on vast corpora have gained prominence, showing impressive capability in zero-shot and few-shot learning. Since the training dataset of LLMs is excessively large (for example, approximately 400 billion BPE [Sennrich et al., 2016] tokens for GPT-3), the required compute for training LLMs is also significantly increased compared to previous PLMs such as GPT, BERT and BART.

Therefore, such models are usually addressed as Large Language Models (LLMs). In this section, we will introduce the most representative LLMs: GPT-3 and its most influential successor - InstructGPT (the base model of ChatGPT).

2.2.1 GPT-3: Language Models are Zero-shot Learners

Following the autoregressive generation pre-training objective of GPT and GPT-2, OpenAI pushes the boundary of generative pre-training even furthe, Brown et al. [2020] proposed GPT-3, a much larger and more powerful autoregressive language model consisting of 175 billion parameters (approximately 100× larger than GPT-large) that is pre-trained on a very large collection of texts including CommonCrawl [2], internet books, WebText and English Wikipedia [Radford et al., 2019, Raffel et al., 2020, Kaplan et al., 2020], which is filtered and then mixed to form the training dataset for GPT-3.

GPT-3 demonstrates impressive zero-shot learning capabilities, i.e. it can generalize to new tasks without requiring fine-tuning or additional training data. This is achieved by leveraging its vast knowledge acquired during the pre-training phase. GPT-3 has been shown to outperform state-of-the-art models on several natural language processing tasks, such as machine translation, question-answering, and summarization Brown et al. [2020]. One of the main innovations in GPT-3 is the use of in-context learning. The model leverages the context provided in the input prompt to guide its response generation, enabling it to adapt its behavior to a wide range of tasks. This flexibility allows GPT-3 to perform tasks like translation, summarization, and even simple programming tasks, all without explicit task-specific training. Despite its impressive performance, GPT-3 has some limitations. Its large size makes it computationally expensive to train and deploy, and its autoregressive nature can lead to errors propagating through the generated text. Moreover, the model may produce plausible but incorrect or nonsensical answers, and its behavior can be sensitive to the phrasing of the input prompt.

[2] `https://commoncrawl.org/the-data/`

2.2.2 InstructGPT: Training language models to follow instructions with human feedback

InstructGPT is a language model that is designed to follow instructions provided in the input prompt and generate useful outputs [Ouyang et al., 2022]. It builds on top of GPT-3 [Brown et al., 2020] and GPT-CodeX [Chen et al., 2021]. The primary difference between InstructGPT and early GPT models is the introduction of human feedback during the training process, which allows the model to learn from human demonstrations and comparisons.

The training process of InstructGPT consists of two main steps: pre-training and fine-tuning. During pre-training, the model is trained on a large corpus of text, similar to GPT-3. This enables the model to acquire general language understanding and knowledge. In the fine-tuning (instruction tuning [Wei et al., 2022]) stage, the model is fine-tuned using a dataset that consists of input-output (instruction-response) pairs, where each pair represents an instruction and its corresponding desired output that is annotated by humans.

To incorporate human feedback, a reward model is trained using a dataset of comparisons. These comparisons consist of two alternative responses for a given instruction ranked by quality (which one is preferred). The reward model, R_θ, is trained to maximize the reward score of the preferred response given an instruction, where θ represents the model parameters. The objective function for the reward model is as follows:

$$J(\theta) = \frac{1}{N} \sum_i \sum_j \sum_k log(R_\theta(x_i, y_j) - R_\theta(x_i, y_k)) \tag{2.17}$$

where, x_i represents the instruction, y_j and y_k are the two alternative responses respectively. y_j is the preferred response compared to y_k. The overall training objective is to maximize the scalar reward score of y_j (the preferred response) over y_k.

After training the reward model R_θ, the language model is then fine-tuned

based on R_θ using Proximal Policy Optimization (PPO) Schulman et al. [2017], an algorithm designed for reinforcement learning, the optimization objective is as follows:

$$J(\varphi) = \frac{1}{M}\sum_{m}(R_\theta(x_m, y_m) - \beta log\frac{\mathcal{A}_\varphi^{\text{PPO}}(y_m|x_m)}{\mathcal{A}_\varphi^{\text{SFT}}(y_m|x_m)}) + \gamma\frac{1}{K}\sum_{k}(log(\mathcal{A}_\varphi^{\text{PPO}}(x_k))) \quad (2.18)$$

where φ represents the parameters of the language model $\mathcal{A}_\varphi$, and $\mathcal{A}_\varphi^{\text{PPO}}$ is the language model fine-tuned on $\mathcal{A}_\varphi^{\text{SFT}}$ with PPO and $\mathcal{A}_\varphi^{\text{SFT}}$ is the language model trained with Supervised Fine-tuning. The first term is the loss from the reward model R_θ. The purpose of the second term $log\frac{\mathcal{A}_\varphi^{\text{PPO}}(y_m|x_m)}{\mathcal{A}_\varphi^{\text{SFT}}(y_m|x_m)})$ is to penalize over-optimization of the reward model [Ouyang et al., 2022], in order words - preventing $\mathcal{A}_\varphi^{\text{PPO}}$ from being trained too far away from $\mathcal{A}_\varphi^{\text{SFT}}$. The third term is the pre-training objective of casual language modeling [Radford et al., 2018]. The coefficients β and γ are used to control the degree of penalty term and pre-training objective.

InstructGPT demonstrates improved performance in following instructions and generating useful outputs, thanks to the incorporation of human feedback during training. This approach shows promise for future applications of language models, where the ability to follow instructions accurately is crucial. Furthermore, an enhanced variant of InstructGPT - ChatGPT [3] with multi-turn conversational ability has been developed and released to the public. The presence of ChatGPT has attracted much attention as it shows excellent ability in handling various NLP (or even beyond) tasks such as grammar error correction, reading comprehension, summarization and translation.

[3] `https://openai.com/blog/chatgpt`

2.3 Vision-Language Pre-trained Models: CLIP - Learning Transferable Visual Models From Natural Language Supervision

Radford et al. [2021] proposed CLIP, which is a pre-trained vision-language model that learns transferable visual features from natural language supervision via pre-training on large-scale web image-text corpora with contrastive learning. The main idea behind CLIP is to train a visual model alongside a language model to understand images and text in a multi-modal fashion. This is achieved by optimizing the model to predict which textual description corresponds to a given image, and vice versa.

The CLIP framework consists of two main components: an image encoder and a language encoder. The image encoder, f_θ, is a deep Convolutional Neural Network (CNN) [LeCun et al., 1998, He et al., 2016] or Vision Transformer (ViT) [Dosovitskiy et al., 2020], while the language encoder, g_ϕ, is a transformer-based architecture Vaswani et al. [2017], Devlin et al. [2019a]. Both encoders are trained jointly to align the image representation and the textual description in a common embedding space.

The training objective for CLIP is to maximize the mutual information between the image and text representations. Given an image x and a textual description y, the model computes the similarity between their respective embeddings as follows:

$$s(x, y) = f_\theta(x)^\top g_\phi(y) \tag{2.19}$$

The CLIP model is trained to maximize the similarity between the correct pair of image and text, while minimizing the similarity between incorrect pairs. The softmax function is used to compute the probability of a correct match:

$$p(y|x) = \frac{e^{s(x,y)}}{\sum_{y' \in Y} e^{s(x,y')}} \tag{2.20}$$

$$p(x|y) = \frac{e^{s(x,y)}}{\sum_{x' \in X} e^{s(x',y)}} \tag{2.21}$$

The pre-training objective is to maximize the log-likelihood of the correct image-text pairs while minimizing the log-likelihood of incorrect image-text pairs within the same batch (in-batch negatives [Gao et al., 2021]) :

$$J_{i2t}(\theta, \phi) = -\frac{1}{N} \sum_{i=1}^{N} \frac{p(y_i|x_i)}{\sum_j^N p(y_j|x_i)} \tag{2.22}$$

$$J_{i2t}(\theta, \phi) = -\frac{1}{N} \sum_{i=1}^{N} \frac{p(x_i|y_i)}{\sum_j^N p(x_j|y_i)} \tag{2.23}$$

By optimizing this objective, CLIP learns to align the embeddings of images and their corresponding textual descriptions in a common semantic space. This makes the model capable of generalizing and transferring knowledge across various visual tasks and natural language understanding tasks, such as object classification, image captioning, and visual question answering. More importantly, the alignment of textual descriptions to images have enabled the use of more flexible labels expressed in natural language instead of a fixed set of pre-designed labels, showing superior performance especially in zero-shot setting compared to fully supervised systems on some vision tasks Radford et al. [2021].

CLIP demonstrates strong performance on a wide range of visual and language tasks including image classification, image retrieval and even video-related tasks [Radford et al., 2021, Li et al., 2021, Xu et al., 2021a, Bain et al., 2022, Lei et al., 2022, Li et al., 2022b], often outperforming traditional supervised learning methods that rely on large amounts of task-specific labeled data.

2.4 Fine-tuning, Adaptation and Knowledge Incorporation for Pre-trained Language Models

In the context of incorporating external knowledge into Pre-trained Language Models (PLMs), it is important to understand the concepts of fine-tuning and adaptation. These concepts play an important role in leveraging existing pre-trained models and enhancing their performance in specific tasks.

2.4.1 Fine-tuning and Adaptation

Fine-tuning and adaptation refer to the process of updating the parameters of a pre-trained model using task-specific data [Peters et al., 2018, Devlin et al., 2019a, Radford et al., 2018]. During the fine-tuning phase, the pre-trained model's parameters are updated based on the task-specific data, enabling it to learn task-specific patterns and improve performance on the target task [Sun et al., 2019a, Liu et al., 2019a, Dodge et al., 2020, Chen et al., 2020a, Yang and Ma, 2022, Chiang and Lee, 2022]. Fine-tuning is an effective way to transfer the knowledge learned during pre-training to new tasks, as it allows the model to leverage its pre-existing knowledge while adjusting to the specific requirements of the targeted downstream task [Radford et al., 2019, Joshi et al., 2020, Beltagy et al., 2020]. Additionally, there are more efficient fine-tuning approaches that optimize a pre-trained model for a specific downstream task by updating only a small group of task-specific parameters while retaining the original pre-trained models [Ding et al., 2023], such as prefix-tuning [Li and Liang, 2021].

By understanding the concepts of fine-tuning and adaptation, we can effectively leverage PLMs and enhance their performance in specific tasks by utilizing task-specific data and making architectural modifications for the purpose of incorporating external knowledge.

2.4.2 Incorporating Knowledge into Pre-trained Language Models

Although the contextualized word representations learned by large-scale PLMs encode rich syntactic and semantic information [Jawahar et al., 2019, Clark et al., 2019, Tenney et al., 2019], they still lack certain knowledge such as world knowledge from knowledge graphs, factual knowledge, and commonsense knowledge that maybe important for certain tasks, especially knowledge-intensive tasks. For example, although models like BERT can capture the co-occurrences among *Apple*, *Tim Cook*, *CEO*, they cannot establish explicit connections that *Tim Cook is the CEO of Apple*. Such knowledge needs to be explicitly injected into pre-trained models [Zhang et al., 2019]. Also, pre-trained models lack factual knowledge. Taking BERT as an example, if we mask *CEO* and substitute *Apple* with *Microsoft* in *Tim Cook is the CEO of Apple*, the resulting sentence is *Tim Cook is the [MASK] of Microsoft*. The masked token predicted by BERT is *CEO* with a high probability. Moreover, pre-trained models lack commonsense knowledge. For instance they cannot detect that *how many eyes does the Earth have?* is an nonsensical question. Such knowledge cannot be captured through self-supervised pre-training on text corpus, supervisory signals from an external knowledge base are needed for Pre-trained Language Models.

Various approaches have been investigated and employed to incorporate knowledge into PLMs [Sun et al., 2019b, Zhang et al., 2019, Peters et al., 2019, Yu et al., 2020, Qiu et al., 2020, Roy and Pan, 2020, He et al., 2020, Colon-Hernandez et al., 2021, Lyu et al., 2020b, Wang et al., 2021c, Wei et al., 2021b]. Most of them focus on injecting structured knowledge. We will present two examples: ERNIE – incorporating entity knowledge in pre-training stage – and K-BERT – injecting domain-specific knowledge information in the fine-tuning and inference phases.

2.4.2.1 ERNIE: Incorporating entity knowledge into language models

In order to enrich text representations with informative entities for better language understanding, Zhang et al. [2019] propose to inject entity information from an

external knowledge base into Pre-trained Language Models. The proposed model ERNIE comprises a text encoder and a knowledge encoder. The text encoder, which is adopted from BERT, is used to encode the text. The knowledge encoder, which is the proposed key component, is responsible for fusing entity representations and textual representations. The objective of ERNIE is to randomly mask aligned *<word,entity>* pairs (e.g. by masking *<entity>* there will be no entity information fused into the representations of *<word>*) then train ERNIE model to predict the masked entity based on the fused representations.

ERNIE is pre-trained on English Wikipedia containing 4500 million subwords [Johnson et al., 2017, Kudo and Richardson, 2018], and the entity embeddings are obtained from Wikidata using TransE [Bordes et al., 2013]. The experimental results show that ERNIE outperforms BERT on knowledge-rich tasks including relation classification and entity typing. ERNIE also obtains comparable performance with BERT on other common NLP tasks, demonstrating the efficacy of the knowledge fusion approach.

Specifically, suppose a sequence of words $(w_1, w_2, ..., w_n)$ and a sequence of entity tokens $(e_1, e_2, ..., e_m)$ that are aligned with words in $(w_1, w_2, ..., w_n)$. Firstly, the text encoder will encode the word sequence $(w_1, w_2, ..., w_n)$ to generate its representations $H = \{h_1, h_2, ..., h_n\}$. Then the text representations $H = \{h_1, h_2, ..., h_n\}$ and the representations of entities $E = (e_1, e_2, ..., e_m)$[4], which are obtained from pre-trained entity embedding, TransE [Bordes et al., 2013], will be fed into the knowledge encoder where H and E are fused. For h_i supposing its aligned entity token in E is e_j, then w_i and e_j will be updated by:

$$d_{ij}^k = g_d(W_d^k[h_i^{k-1}, e_j^{k-1}])), h_i^k = g_h(W_h^k d_{ij}^k), e_j^k = g_e(W_e^k d_{ij}^k) \tag{2.24}$$

where k represents the k-th layer of the knowledge encoder that is composed of L layers, d_{ij}^k is the fused representations of word h_i^{k-1} and its aligned entity e_j^{k-1} which are then updated and assigned as the new values of h_i^k and e_j^k, and g_d, g_h, g_e are

[4]For notational simplicity we still use $(e_1, e_2, ..., e_m)$ to represent entity embeddings

corresponding activation functions. After the knowledge fusion stage, the generated text representations are $\tilde{H} = \{h_1^L, h_2^L, ..., h_n^L\}$.

The new objective of ERNIE is to randomly mask the aligned *<word,entity>* pairs (e.g. by masking e_j in $<h_i, e_j>$ there will be no entity information being fused into h_i^k) then train ERNIE model to predict the masked entity based on $\tilde{H} = \{h_1^L, h_2^L, ..., h_n^L\}$:

$$J_{entity}(\theta) = \sum_E \sum_{i,j} logP(< w_i, e_j > |h_1^L, h_2^L, ..., h_n^L) \tag{2.25}$$

where the probability of predicting an entity e_j for word w_i is modeled by:

$$P(< w_i, e_j > |h_1^L, h_2^L, ..., h_n^L) = \frac{e^{\psi(h_i^L, e_j^L)}}{\sum_t \psi(h_i^L, e_t^L)} \tag{2.26}$$

Note that in ERNIE the probability distribution is normalized over the entity list $(e_1, e_2, ..., e_m)$ not the whole entity vocabulary for computational efficiency. Similar to the practice of the masked token prediction in BERT, Zhang et al. [2019] propose a masking strategy for a word-entity pair $<h_i, e_j>$ (i). 5% of the time, e_j will be replaced with another randomly sampled entity $e_{\hat{j}}$ (ii). 15% of the time, e_j will be masked (iii). 80% of the time, e_j will stay unchanged. Moreover, ERNIE also uses the masked token prediction $J_1(\theta)$ and next sentence prediction $J_2(\theta)$ as pre-training objectives, therefore the overall objective of ERNIE is the sum of the three objectives above: $J(\theta) = J_{entity}(\theta) + J_1(\theta) + J_2(\theta)$. By using such pre-training objectives the text representations produced by ERNIE are expected to contain not only the semantic patterns of words but also the entity information obtained from knowledge fusion.

2.4.2.2 K-BERT: Injecting knowledge graph into BERT for enhanced language representations

Different from Zhang et al. [2019] where the entity knowledge is injected during the pre-training phase, Liu et al. [2020a] propose to incorporate knowledge in the

fine-tuning and inference phases by explicitly injecting knowledge graph information into text sequences. Their aim in doing this is to reduce the required computational resources for pre-training and knowledge graph embeddings. K-BERT firstly uses the entity information in sequence $(w_1, w_2, ..., w_n)$ to obtain the relevant $< entity_1, relation, entity_2 >$ triples from a knowledge graph, then these triples are injected into the original sequence directly by appending $< relation, entity_2 >$ to $< entity_1 >$ in the sequence. For examples, if a triple $< Bill\_Gates, CEO_{of}, Microsoft >$ is retrieved for sentence *Bill Gates calls for 'Green industrial revolution' to beat climate crisis*, then K-BERT will inject the triple into the sentence by modifying it to: *Bill Gates CEO of Microsoft calls for 'Green industrial revolution' to beat climate crisis*. It is worth noting that although *CEO of Microsoft* is inserted between *Bill Gates* and *calls for ...*, *CEO of Microsoft* still shares the same position embeddings [Vaswani et al., 2017] with *calls for ...*. In other words, the injection of *CEO of Microsoft* won't affect the original order of the sentence. After the injection of entity triples, the modified sequence can be fed into the transformer encoders.

Note that the design of K-BERT enables the incorporation of any domain knowledge graph for specific tasks without pre-training and knowledge embeddings since the entity and relation information can be directly injected into the text sequence. Therefore when employing K-BERT for specific tasks, one should use the same pre-training objectives as BERT [Devlin et al., 2019a] or directly initialize the transformer encoders using a public Google BERT model [Devlin et al., 2019a] then use appropriate knowledge graphs in the fine-tuning stage to inject entity and relation information into the text sequences and train K-BERT with task-specific objectives. In the experiments of Liu et al. [2020a], K-BERT is pre-trained on Chinese corpora including WikiZh and WebtextZh, and the knowledge graphs used in downstream tasks include CN-DBpedia [Xu et al., 2017], HowNet [Dong et al., 2010] and MedicalKG. Experimental results show K-BERT outperforms vanilla BERT on text classification tasks for the e-commerce domain, XNLI [Conneau et al., 2018]

and domain-specific NER.

2.5 Document-level Sentiment Analysis with User and Product Context

Document-level sentiment analysis aims to predict the sentiment polarity of text usually taking the form of a lengthy document. Sentiment analysis with user and product information [Tang et al., 2015b] is a particular kind of sentiment classification task in which the corresponding user (or review author) and the product that was evaluated are taken into consideration when predicting sentiment polarity.

Neural networks have been widely used in sentiment classification [Socher et al., 2013, Kim, 2014, dos Santos and Gatti, 2014, Tang et al., 2015a, Wang et al., 2016]. Most existing work focuses solely on the text itself, with Tang et al. [2015b] being the first to identify the importance of incorporating user and product information for sentiment classification. In their work, a CNN-based encoder was used to obtain document representations, and user and product information are represented in both vector and matrix form, and injected in the word embedding layer and classification layer. Subsequently, many methods were proposed for the purpose of better capturing user preferences and product-specific sentiment. Chen et al. [2016b] used an LSTM encoder to obtain the sentence-level representation before combining them into a document representation. In their model, user and products are represented as vectors in the embedding matrix and user-product vectors are used to gather important information at both word and sentence-level through an attention mechanism. Ma et al. [2017] proposed a cascading multi-way attention to model the dependencies among user, product and review. Dou [2017], Long et al. [2018] adopted a memory network to capture user and product information. To alleviate the cold-start problems Amplayo et al. [2018] proposed to utilize similar user information when a given user's reviews are limited. Furthermore, Amplayo [2019] proposed a novel model that represents users and products as chunk-wise importance weight matrices in order

to improve the performance by reducing the number of parameters to be optimized. Zhang et al. [2021b] proposed a multi-attribute encoder using bilinear projections between attributes and texts on top of BERT [Devlin et al., 2019a] to make better use of attribute (user and product) information.

Most aforementioned studies focus on capturing user and product preferences. However they neglect to incorporate the reviews from the same user and product. We hypothesize that explicitly utilizing such extra context is helpful for sentiment classification. We explore this further in Chapter 3

2.6 Semantic Role Labeling

Semantic Role Labeling (SRL) is a NLP task that aims to reveal the underlying semantic structure of a sentence by identifying predicate-argument structure and classifying their semantic roles. This process is important for understanding the meaning of natural language sentences and plays a useful role in various NLP tasks such as information extraction and machine translation systems [Liu and Gildea, 2010, Christensen et al., 2011].

In SRL, predicates are typically verbs that represent an action or a state, and arguments are phrases representing roles or entities taking part in that action or state. The semantic roles captured by SRL are defined in frameworks such as PropBank [Kingsbury and Palmer, 2002, Palmer et al., 2005] and FrameNet [Baker et al., 1998]. Frameworks such as Propbank use arguments like *Arg0, Arg1, Arg2* and *ArgM*, to represent different arguments in a sentence. For example, in sentence *John ate an apple in the classroom.*, the verb *ate* serves as a predicate that conveys the action *eating*. The other arguments in this sentence should be tagged as:

- *Arg0*: John - The agent who performs the action
- *Arg1*: an apple - The object that undergoes the action performed
- *ArgM-Loc*: in the classroom - The location where the action takes place

Traditional methods for SRL typically use the syntactic structure of the sentence and develop hand-crafted rules or employ supervised machine learning techniques like Support Vector Machines (SVM) [Park et al., 2004]. With the introduction of deep learning techniques, SRL systems mainly focus on using neural networks such as Long Short-Term Memory (LSTM) networks [He et al., 2017], and more recently, PLMs like BERT [Devlin et al., 2019a, Shi and Lin, 2019] are employed for improved performance.

Since SRL is capable of producing structured representations of natural language sentences, so incorporating the knowledge of SRL into PLMs can potentially improve their performance by providing semantic information that can inform higher-level reasoning and facilitate machine understanding of natural language. We will demonstrate approaches incorporating the knowledge of SRL into PLMs in Chapter 4 and Chapter 5.

2.7 Unsupervised Question Answering and Question Generation

Question Generation (QG) aims to generate plausible questions according to a given passage and answer pair. For example, given the passage *"The Eiffel Tower is a wrought-iron lattice tower located in Paris, France. It was completed in 1889 and is named after the engineer Gustave Eiffel."* and the answer *"Gustave Eiffel"*, the QG model would generate a question such as *"Who is the engineer that the Eiffel Tower in Paris is named after?"*. Traditional approaches to QG mostly employ linguistic templates and rules to transform declarative sentences into interrogatives [Heilman and Smith, 2009]. Recently, Dhole and Manning [2020] showed that, with the help of advanced neural syntactic parsers, template-based methods are capable of generating high-quality questions from texts.

Neural seq2seq generation models have additionally been widely employed in QG, with QG data usually borrowed from existing QA datasets [Du et al., 2017,

Sun et al., 2018, Ma et al., 2020]. Furthermore, reinforcement learning has been employed by Zhang and Bansal [2019], Chen et al. [2019], Xie et al. [2020] to directly optimize discrete evaluation metrics such as BLEU [Papineni et al., 2002]. Lewis et al. [2020] and Song et al. [2019] show that a large-scale pre-trained model can achieve state-of-the-art performance for supervised QG [Dong et al., 2019, Narayan et al., 2020].

BLEU [Papineni et al., 2002], ROUGE [Lin, 2004] and Meteor [Banerjee and Lavie, 2005] metrics are commonly borrowed from text generation tasks to evaluate QG where the system-generated question is compared with the ground-truth question based on n-gram overlap. Even with respect to original text generation tasks, however, the use of such metrics has been questioned [Callison-Burch et al., 2006, Reiter, 2018]. Such metrics are particularly problematic for QG evaluation since multiple plausible questions exist for a given passage and answer. Consequently, there has been a shift in focus to evaluating QG using an extrinsic evaluation that generates synthetic QA pairs for the purpose of evaluating their effectiveness as a data augmentation or unsupervised QA approach [Alberti et al., 2019, Puri et al., 2020, Shakeri et al., 2020].

In unsupervised QA, the QA model is trained using synthetic data based on a QG model instead of an existing QA dataset. In order to train the QG systems used to generate synthetic QA data, various approaches have been proposed: 1) Alberti et al. [2019], Puri et al. [2020], Shakeri et al. [2020] additionally employ existing QA datasets to train a QG model where the passage and answer serve as the input and question is used as output target. 2) Instead of resorting to existing QA datasets, unsupervised QG methods have been employed, such as Unsupervised Neural Machine Translation [Lewis et al., 2019]. Fabbri et al. [2020], Li et al. [2020] propose template/rule-based methods for generating questions and employ retrieved paragraphs and cited passages as source passages to alleviate the problems of lexical similarities between passages and questions. Those approaches either rely on existing annotated QA datasets or suffer from low-quality generated questions, we will explore

this problem in Chapter 4.

2.8 Multi-modal Video Question Answering

Video Question Answering (VideoQA) [Yang et al., 2003, Tapaswi et al., 2016, Zhao et al., 2017, Kim et al., 2020, Xiao et al., 2022a] aims to answer a textual question based on a video where the question is about the understanding of the video content and the answer can either be selected from a candidate set or be generated. VideoQA is a complex and challenging task that requires a deep understanding of the spatio-temporal nature of videos and the ability to reason about objects, relations, and events across visual and linguistic domains [Lei et al., 2018, Yun et al., 2021, Xiao et al., 2022a]. To tackle this task, existing research has focused on cross-modal interaction with the aim of understanding videos under the guidance of questions. For example, Visual Relation Grounding in Videos (vRGV) proposed by Xiao et al. [2020], have addressed the challenges of spatio-temporal localization and the dynamic nature of visual relations in videos. Hierarchical Object-oriented Spatio-Temporal Reasoning (HOSTR) networks [Dang et al., 2021] focus on object-oriented reasoning, maintaining consistent object lifelines within a hierarchically nested spatio-temporal graph. Invariant Grounding for VideoQA [Li et al., 2022c] is another learning framework that focuses on grounding question-critical scenes and improving reasoning abilities by shielding the answering process from the negative influence of spurious correlations. There has also been a shift towards modeling video as a conditional graph hierarchy [Xiao et al., 2022a], which aligns with the multi-granular essence of linguistic concepts in language queries and improves performance and generalization across different types of questions. These methods have collectively contributed to advances in the field of VideoQA, enhancing accuracy, visual explainability, and generalization ability across various tasks and datasets.

However, existing approaches often have limitations, such as the problem of generally fail to take into account the explicit semantic connection between questions

and the corresponding visual information at the event level and aligning audio and visual features in the same vector space with respect to the semantic-level information, especially when the features have different scales and granularities [Zhong et al., 2022a, Gan et al., 2022]. We atttempt to address these limitations in our work described in Chapter 5.

Chapter 3

User-Product Context for Sentiment Analysis

In this chapter, we will discuss how to utilize user and product context for document-level Sentiment Analysis. We present two novel approaches explicitly making use of the textual information in the historical reviews associated with each specific user and product to improve document-level Sentiment Analysis. This chapter is based on two papers published at *COLING 2020 - Improving Document-Level Sentiment Analysis with User and Product Context* [Lyu et al., 2020a] and the findings of *ACL 2023 - Exploiting Rich Textual User-Product Context for Improving Personalised Sentiment Analysis* [Lyu et al., 2023d].

3.1 Sentiment Analysis with User and Product Context

Document-level sentiment analysis aims to predict sentiment polarity of text that often takes the form of product or service reviews. Tang et al. [2015b] demonstrated that modelling the individual who has written the review, as well as the product being reviewed, is worthwhile for polarity prediction, and this has led to exploratory work on how best to combine review text with user/product information in a neural architecture [Chen et al., 2016b, Ma et al., 2017, Dou, 2017, Long et al., 2018, Amplayo, 2019, Amplayo et al., 2018]. A feature common amongst past studies is that user and product IDs are modelled as embedding vectors whose parameters are learned during training. We take this idea a step further and represent users and

products using the *text of all the reviews belonging to a single user or product* - see Figure 3.1.

There are two reasons to incorporate review text into user/product modelling. Firstly, the reviews from a given user will reflect their word choices when conveying sentiment. For example, a typical user might use words such as *fantastic* or *excellent* with correspondingly high ratings but another user could use the same words sarcastically with a low rating. Similarly, a group of users writing a review of the same product may use the same or similar opinionated words to refer to that product. Secondly, learning meaningful user and product embeddings that are only updated by backpropagation is challenging when a user or product has a limited number of reviews. However, even with a small number of reviews, it is still possible to extract useful information from the text. For example, in the case of a specific user or product with only a few reviews, their corresponding embeddings will only be updated several times during the training process, resulting in sub-optimal representations. On the other hand, the textual information in their historical reviews is more useful in reflecting their rating preferences. We present two methods for using the text of historical reviews. The first approach is presented in Chapter 3.2 and the second is presented in Chapter 3.3.

We compare performance with a range of systems and results show that our approach works, improving on state-of-the-art results for all three benchmark datasets (IMDB, Yelp-13 and Yelp-14).[1] We also compare to a version of our own system which does not use the review text representations to encode user and product information. While it performs competitively with other systems, demonstrating the efficacy of our basic architecture, it does not work as well as our proposed system, particularly for reviews written by users or products with only a small number of reviews.

[1] http://ir.hit.edu.cn/~dytang/paper/acl2015/dataset.7z

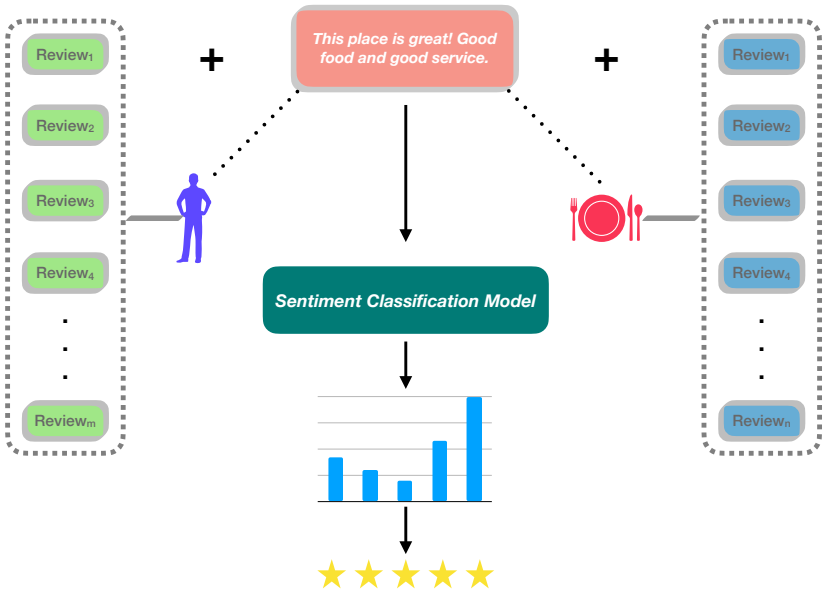

Figure 3.1: Utilizing all historical reviews of corresponding user and products.

3.2 Method-1

A naive approach might compute representations of all the reviews of a given user or product each time we have a new training sample but this would be too expensive, and we instead propose the following incremental approach: With each new training sample, we obtain the review text representation, with BERT [Devlin et al., 2019a] as our encoder, before using the representation together with user and product vectors to obtain a user-biased document representation and a product-biased document representation, which are then employed to obtain sentiment polarity. We then add the user-biased and product-biased document representations to the corresponding user and product vectors, so that they are ready for the next sample. In doing so, we incrementally store and update representations of reviews for a given user and product. Unlike Ma et al. [2017], who use a hierarchical structure in which sentence representations are first computed before being combined into a document

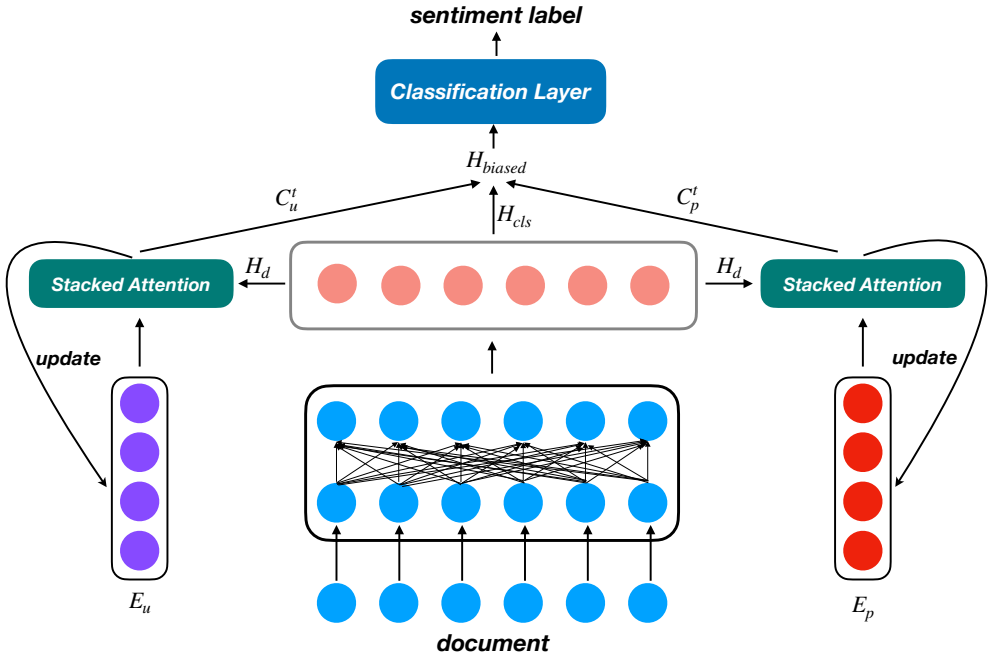

Figure 3.2: Overall architecture of our model, where E_u and E_p are user and product representations.

representation, We let the words in the text directly attend to each other. The architecture we propose is depicted in Figure 3.2 and is explained in more detail in Section 3.2.1.

3.2.1 Methodology

An overview of our model architecture is shown in Figure 3.1. The input to our model consists of d, u, p, which are the document, the user id and the product id respectively. u and p are both mapped to embedding vectors, $E_u, E_p \in \mathbf{R^h}$. d is fed into the BERT encoder to generate a document representation $H_d \in \mathbf{R^{L \times h}}$ where L is the length of document after tokenization. We then inject E_u and E_p, to get the user-product biased document representation $H_{biased} \in \mathbf{R^h}$. Finally, we feed the biased document representation H_{biased} into a linear layer followed by a *softmax* layer to get the distribution of the sentiment label y. We use *cross-entropy* to calculate the loss between the predictions and ground-truth labels.

Injecting user and product preferences We adopt stacked **multi-head-attention** (Q, K, V) [Vaswani et al., 2017] to model the connections between the current document and user/product vectors, which correspond to all historical reviews composed by the user or about the product to date. In a typical dot-product attention (Q, K, V), $Q \in R^{L_Q \times h}$, $K \in R^{L_K \times h}$, $V \in R^{L_V \times h}$. Generally, $L_K = L_v$. E_u and E_p are regarded as queries, H_d as keys and values. We compute the user-specific document representation, C_u^t, and product-specific document representation, C_p^t as follows:

$$C_u^t = \textbf{stacked-attention}(E_u, H_d, H_d) \quad C_p^t = \textbf{stacked-attention}(E_p, H_d, H_d) \tag{3.1}$$

where $C_u^t = attention(C_u^{t-1})$, $C_u^0 = E_u$ (similarly for C_p^t), and t is the number of layers of the attention function. In Equation (3.1), $C_u^t \in R^h, C_p^t \in R^h$.

We adopt a *gating mechanism* to obtain importance vectors, z_u and z_p, to control the *contribution* of user-specific and product-specific document representations to the output classification:

$$z_u = \sigma(W_{zu} C_u^t + W_{zh} H_d + b_u) \quad z_p = \sigma(W_{zp} C_p^t + W_{zh} H_d + b_p) \tag{3.2}$$

Finally, we obtain the biased document representation H_{biased} by:

$$H_{biased} = H_{cls} + z_u \odot C_u^t + z_p \odot C_p^t \tag{3.3}$$

where $H_{cls} \in \mathbf{R^h}$ is the final hidden vector of the [CLS] token (which is a special token that is added at the beginning of texts used for classification) [Devlin et al., 2019a] and $\odot$ is element-wise product.

Updating the user and product matrix To implement our idea of using all reviews composed by u and all reviews about p, we incrementally add the current

user/product-specific document representation to the corresponding entries in the embedding matrix at each step during training:

$$E_u^{'} = \sigma(E_u + \lambda_u C_u^t) \quad E_p^{'} = \sigma(E_p + \lambda_p C_p^t) \tag{3.4}$$

where λ_u and λ_p are both learnable real numbers that control the degree to which the representation of the current document should be employed.

With E_u being updated every step during training process, it can *memorize* all reviews attached to the corresponding user, the same for E_p. Furthermore, we apply two linear transformations to user and product vectors. The first linear layer is used to transform user and product vectors to the same dimension as the document representation, the second layer is used to transform them back into the original dimension:

$$E_u = W_{in} U_u + b_{in} \quad E_p = W_{in} P_p + b_{in} \tag{3.5}$$

$$U_u^{'} = W_{out} E_u^{'} + b_{out} \quad P_p^{'} = W_{out} E_p^{'} + b_{out} \tag{3.6}$$

where $U \in \mathbf{R}^{\mathbf{M} \times \mathbf{h}^{'}}$ is the user embedding matrix, $P \in \mathbf{R}^{\mathbf{N} \times \mathbf{h}^{'}}$ is the product embedding matrix, M and N are the total number of user and product respectively, U_u and P_p are rows in U and P corresponding to u and p. In Equations (5) and (6), $W_{in} \in \mathbf{R}^{\mathbf{h} \times \mathbf{h}^{'}}, b_{in} \in \mathbf{R}^{\mathbf{h}}$ and $W_{out} \in \mathbf{R}^{\mathbf{h}^{'} \times \mathbf{h}}, b_{out} \in \mathbf{R}^{\mathbf{h}^{'}}$. $h^{'}$ is the embedding size, generally $h^{'} \leq h$.

Objective function We use *Cross-Entropy* function to calculate the loss between the predictions of our model and ground-truth labels:

$$Loss = -\sum_{i=1}^{n} \sum_{j=1}^{m} y_{i,j} log(p(y_{i,j} | d_i, u_i, p_i)) \tag{3.7}$$

where n is the number of samples and m is the number of all classes, $y_{i,j}$ represents

the actual probability of the i-th sample belonging to $class_j$, $y_{i,j}$ is 1 only if the i-th sample belongs to $class_j$ otherwise it's 0. $p(y_{i,j}|d_i, u_i, p_i)$ is the probability the i-th sample belongs to $class_j$ predicted by our model.

3.2.2 Experimental Setup

Our experiments are conducted on the IMDB, Yelp-13 and Yelp-14 benchmark datasets where the task is to predict the fine-grained sentiment polarity (5 classes for Yelp and 10 classes for IMDB) from the review texts, statistics of which are shown in Table 3.1. We use the *BERT-base* model from HuggingFace [Wolf et al., 2019]. We train our model with a learning rate chosen from {8e-6, 3e-5, 5e-5}, and a weight decay rate chosen from {0, 1e-1, 1e-2, 1e-3}, the optimizer we use is AdamW[Loshchilov and Hutter, 2019]. In our experiments, the number of attention layers t is set to 5. The maximum sequence length to BERT is 512. We select the hyper-parameters achieving the best results on the dev set for evaluation on the test set. Evaluation metrics (Accuracy and RMSE) are calculated using scripts from Scikit-learn [Pedregosa et al., 2011].[2]

Datasets	Classes	Documents	Users	Products	Docs/User	Docs/Product	Words/Doc
IMDB	1–10	84,919	1,310	1,635	64.82	51.94	394.6
Yelp-2013	1–5	78,966	1,631	1,633	48.42	48.36	189.3
Yelp-2014	1–5	231,163	4,818	4,194	47.97	55.11	196.9

Table 3.1: Statistics of IMDB, Yelp-2013 and Yelp-2014.

Datasets	Train	Dev	Test	Words/Doc
IMDB	67,426	8,381	9,112	394.6
Yelp-2013	62,522	7,773	8,671	189.3
Yelp-2014	183,019	22,745	25,399	196.9

Table 3.2: Number of documents per split and average doc length of IMDb, Yelp-2013 and Yelp-2014.

[2]https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Datasets	Users	Products	Docs/User	Docs/Product
IMDB	1,310	1,635	64.82	51.94
Yelp-2013	1,631	1,633	48.42	48.36
Yelp-2014	4,818	4,194	47.97	55.11

Table 3.3: Number of users and products with average amount of documents for each user and product in IMDb, Yelp-2013 and Yelp-2014.

3.2.3 Datasets

Our experiments are conducted on three benchmark English document-level sentiment analysis datasets: IMDb, Yelp-13 and Yelp-14 [Tang et al., 2015b]. Statistics of the three datasets are shown in Table 3.2. The IMDb dataset has the longest documents with an average length of approximately 395 words. All three are fine-grained sentiment analysis datasets: Yelp-2013 and Yelp-2014 have 5 classes, IMDb has 10 classes. Each review is accompanied by its corresponding anonymized user ID and product ID. The average number of reviews for each user/product is shown in Table 3.3.

3.2.4 Results

Our experimental results are shown in Table 3.4. Our proposed model is named IUPC (**I**ncorporating **U**ser-**P**roduct **C**ontext). The first two rows are baseline models: BERT VANILLA which is the basic BERT model without user and product information, i.e. only review text, and IUPC W/O UPDATE, which is the same as our proposed model except that we do not update the user and product embedding matrix by incrementally adding the new review representations. The third row shows our proposed model. We also compare with results from the NLP-progress leaderboard[3] of the following models:

[3]http://nlpprogress.com/english/sentiment_analysis.html

CHIM [Amplayo, 2019] adopts a chunk-wise matrix representation for user/product attributes; injects user/product information in different locations.

CMA [Ma et al., 2017] A hierarchical LSTM encoding the document; injects user and product information hierarchically.

DUPMN [Long et al., 2018] encodes the document using a hierarchical LSTM; adopts two memory networks, one for user information and another for product information.

HCSC [Amplayo et al., 2018] A combination of CNN and Bi-LSTM as the document encoder; injects user/product information with bias-attention.

HUAPA [Wu et al., 2018] adopts two hierarchical models to get user and product specific document representations respectively.

NSC [Chen et al., 2016b] A hierarchical LSTM encoder incorporating user/ product attributes with word and sentence-level attention.

RRP-UPM [Yuan et al., 2019] uses two memory networks besides the user/product embeddings to get refined representations for user/product information.

UPDMN [Dou, 2017] An LSTM model encoding the document; a memory network capturing user/product information.

UPNN [Tang et al., 2015b] adopts a CNN-based encoder and injects user/product information in the embedding and classification layers.

	IMDB		Yelp-2013		Yelp-2014	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
BERT VANILLA	$47.9_{0.46}$	$1.243_{0.019}$	$67.2_{0.46}$	$0.647_{0.011}$	$67.5_{0.71}$	$0.621_{0.012}$
IUPC W/O UPDATE	$52.1_{0.31}$	$1.194_{0.010}$	$69.7_{0.37}$	$0.605_{0.007}$	$70.0_{0.29}$	$0.601_{0.007}$
IUPC (our model)	$53.8_{0.57}$	$\mathbf{1.151_{0.013}}$	$\mathbf{70.5_{0.29}}$	$\mathbf{0.589_{0.004}}$	$\mathbf{71.2_{0.26}}$	$\mathbf{0.592_{0.008}}$
UPNN	43.5	1.602	59.6	0.784	60.8	0.764
UPDMN	46.5	1.351	63.9	0.662	61.3	0.720
NSC	53.3	1.281	65.0	0.692	66.7	0.654
CMA	54.0	1.191	66.3	0.677	67.6	0.637
DUPMN	53.9	1.279	66.2	0.667	67.6	0.639
HCSC	54.2	1.213	65.7	0.660	67.6	0.639
HUAPA	55.0	1.185	68.3	0.628	68.6	0.626
CHIM	**56.4**	1.161	67.8	0.641	69.2	0.622
RRP-UPM	56.2	1.174	69.0	0.629	69.1	0.621

Table 3.4: Experimental Results on IMDB, Yelp-2013 and Yelp-2014. Following previous work, we use Accuracy (Acc.) and Root Mean Square Error (RMSE) for evaluation. There are 10 classes in IMDB and 5 classes in Yelp 2013 and Yelp 2014. We run BERT VANILLA, IUPC W/O UPDATE and IUPC five times and report the average Accuracy and RMSE. The subscripts represent standard deviation.

Our model achieves the best classification accuracy and RMSE on Yelp-2013 and Yelp-2014, and the best RMSE on IMDB. It outperforms previous state-of-the-art results by 1.5 accuracy and 0.042 RMSE on Yelp-2013, by 2.1 accuracy and 0.029 RMSE on Yelp-2014, and by 0.01 RMSE on IMDB. Moreover, it outperforms the two baselines, BERT VANILLA and IUPC W/O UPDATE in both classification accuracy and RMSE on all three datasets.

3.2.5 Analysis

We analyse the results for reviews whose user or product do not have many reviews in the training set and compare our model's performance to the IUPC W/O UPDATE baseline for one dataset (Yelp-2013 dev). We select only reviews where the number of reviews by that user or for that product falls below three thresholds: 40%, 60%, 80% (19, 28, 38 reviews respectively), where % stands for the number of reviews for a given user/product relative to the average number of reviews for all users/products. Table 3.5 shows that our model performs better than IUPC W/O UPDATE when there are only a small number of previous reviews available for a given product/user. In other words, when a user or product does not have many reviews, its IUPC W/O UPDATE embedding which is only updated by gradient descent, cannot capture user/product preference as well as our model which explicitly takes advantage of historical review text in its user/product representations.

	40%		60%		80%	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
IUPC W/O UPDATE	63.0	0.608	64.0	0.665	66.8	0.643
IUPC (our model)	**65.7**	**0.585**	**66.8**	**0.649**	**67.9**	**0.631**

Table 3.5: Analysis of three lower-resource scenarios where % denotes a threshold filter corresponding to the proportion of reviews available relative to the average number in the dataset Yelp-2013 (dev).

In order to get a better idea of where there is room for improvement for IUPC, we examine the 43 Yelp-13 dev set cases, where the predicted label differs from the gold label by more than two points. There are a handful of cases of sarcasm,

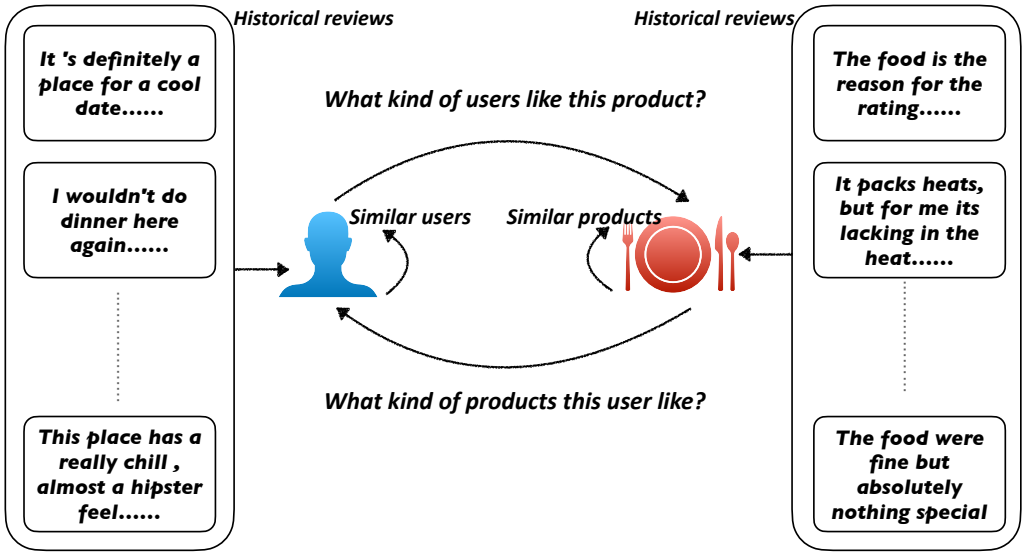

Figure 3.3: Our proposed idea of representing users and products with their historical reviews, which can directly inform user and product preferences, and incorporating the associations between users and products.

e.g. *that* ***lovely*** *tempe waste/tap water taste in the food*, but the most noteworthy phenomenon is mixed sentiment, e.g. *tacos were good the soup was not tasty*, or the more subtle *brave the scary parking and lack of ambiance.* It is not always clear from the reviews which aspect of the service the rating is directed towards. This suggests that aspect-based sentiment analysis [Pontiki et al., 2014] might be useful here, and training an IUPC model for this task is a possible avenue for future work.

3.3 Method-2

We proposed a method that explicitly uses historical reviews in the training process Lyu et al. [2020a]. However, this approach needs to incrementally store review representations during the training process, which results in a more complex model architecture.

As shown in Figure 3.3, we propose an alternative approach. We use pre-trained language models (PLMs) to pre-compute the representations of all historical reviews belonging to the same user/product. Historical review representations are then used

to initialize user/product representations by average pooling over all tokens before again average pooling over all reviews. This allows historical review text to inform the user and product preference, while minimizing time and memory costs. since the representations of historical reviews are average pooled and the pre-computation is one-time.

In addition, we propose a user-product cross-context module, which cooperates with historical representations of users and products to gather sentiment polarity information from the reviews of other users/products. This module interacts on four dimensions: user-to-user, product-to-product, user-to-product and product-to-user. The former two are used to obtain similar user (product) information, which is useful to model user (product) preference especially when a user (product) has limited reviews. The latter two are used to model the product preference of the user (what kind of products do they like and what kind of ratings would they give to similar products?) and user preference associated with a product (what kinds of users like such products and what kinds of ratings would they give to this product?).

We apply our approach to various English PLMs and test on the same three benchmark English datasets used for method-1 – IMDb, Yelp-2013, Yelp-2014. We find that our approach yields consistent improvements across PLMs (BERT, SpanBERT, Longformer) and achieves substantial improvements over previous state-of-the-art models. We also show the superior performance of our approach when the number of reviews for each user is limited.

Our contributions are two effective, cooperative strategies for improving sentiment analysis with user and product information:

1. initializing user and product representations using their historical reviews
2. a user-product cross-context module which cooperates with Contribution 1 to efficiently incorporate textual associations between users and products from a larger context.

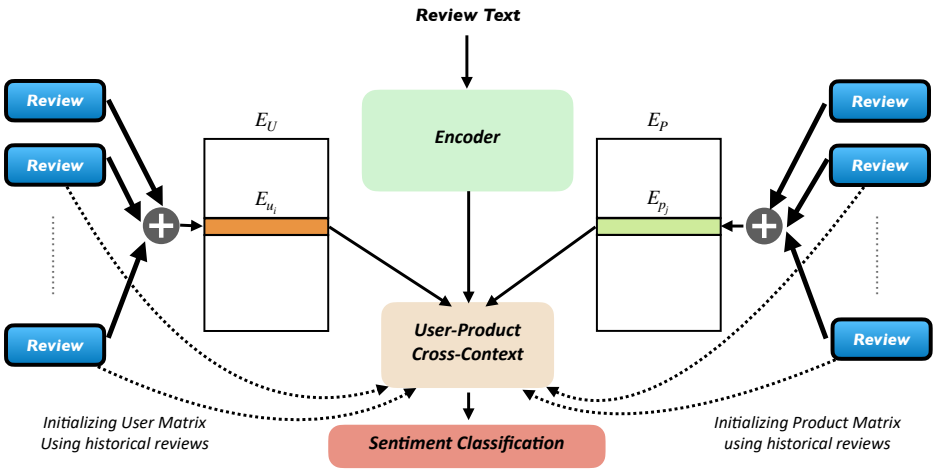

Figure 3.4: Our model architecture. We initialize user representation matrix E_U and product representation matrix E_P. The user vector E_{u_i} and product vector E_{p_j} are fed into user-product cross-context module with document representation H_D. The dashed lines indicate the direct interactions of historical reviews in the cross-context module.

3.3.1 Methodology

An overview of our approach is shown in Figure 3.4. We firstly feed the review text, D, into a PLM encoder to obtain its representation, H_D. H_D is then fed into a user-product cross-context module consisting of multiple attention functions together with the corresponding user embedding vector, u and product embedding vector, p. The output of the user-product module is concatenated with H_D and fed into a linear classification layer to obtain the distribution over all sentiment labels.

Incorporating Textual Information of Historical Reviews For the purpose of making use of the textual information of historical reviews, we initialize all user and product embedding vectors using the representations of their historical reviews. Specifically, assume that we have a set of users $U = \{u_1, ..., u_N\}$ and products $P = \{p_1, ..., p_M\}$. Each user u_i and product p_j have their corresponding historical reviews: $u_i = \{D_1^{u_i}, ..., D_{n_i}^{u_i}\}$ and $p_j = \{D_1^{p_j}, ..., D_{m_j}^{p_j}\}$.

For a certain user u_i, we firstly feed $D_1^{u_i}$ into the transformer encoder to obtain

its representation $H_{D_1}^{u_i} \in \mathbf{R}^{L \times h}$, then we average $H_{D_1}^{u_i}$ along its first dimension:

$$\bar{H}_{D_1}^{u_i} = \frac{\sum H_{D_1}^{u_i}}{T_{D_1}^{u_i}} \tag{3.8}$$

where $\bar{H}_{D_1}^{u_i} \in \mathbf{R}^{1 \times h}$, L is the maximum sequence length, h is the hidden size of the transformer encoder, $T_{D_1}^{u_i}$ is the total number of tokens in $D_1^{u_i}$ excluding special tokens. Therefore, we simply sum the representations of all tokens in $D_1^{u_i}$ and then average it to obtain a document vector $\bar{H}_{D_1}^{u_i}$. The same procedure is used to generate the document vectors of all documents in $u_i = \{D_1^{u_i}, ..., D_{n_i}^{u_i}\}$. Finally, we obtain the representation of u_i by:

$$E_{u_i} = \frac{\sum_{k=1}^{n_i} \bar{H}_{D_k}^{u_i}}{n_i} \tag{3.9}$$

where $E_{u_i} \in \mathbf{R}^{1 \times h}$ is the initial representation of user u_i. The same process is applied to generate the representations of all the other users as well as all products. Finally, we have $E_U \in \mathbf{R}^{N \times h}$ and $E_P \in \mathbf{R}^{M \times h}$ as the user and product embedding matrix respectively. Moreover, in order to control the magnitude of E_U, E_P to prevent it from being too large or too small, we propose scaling heuristics, E_U and E_P:

$$\hat{E_U} = f_U E_U, f_U = \frac{F\_Norm(E)}{F\_Norm(E_U)} \tag{3.10}$$

$$\hat{E_P} = f_P E_P, f_P = \frac{F\_Norm(E)}{F\_Norm(E_P)} \tag{3.11}$$

where F_Norm is Frobenius norm, and E is a normal matrix in which the elements $E_{i,j}$ are drawn from a normal distribution $\mathcal{N}(0, 1)$.

User-Product Information Integration Having enriched user and product representations with historical reviews, we propose a user-product cross-context module for the purpose of garnering sentiment clues from textual associations between users and products. In this module, we adopt four attention operations: *user-to-user, product-to-product, user-to-product* and *product-to-user*. We use *Multi-Head-Attention* [Vaswani et al., 2017] in four attention operations. Specifically, for *Multi-Head-Attention(Q,K,V)*, we use the user representation E_{u_i} or product

representation E_{p_j} as Q and the user matrix E_U and product matrix E_P as K and V. It is important to note that, before using E_{u_i} and E_{p_j}, we fuse the document information $H_{cls} \in \mathbf{R}^{1\times h}$, the representation of the *[CLS]* token, into them as follows:

$$E_{u_i} = g_u(E_{u_i}, H_{cls}), E_{p_j} = g_p(E_{p_j}, H_{cls}), \tag{3.12}$$

where g_u and g_p represent two linear layers combining E_{u_i}/E_{p_j} and H_{cls}.

1. **User-to-User Attention** We use E_{u_i} as the query and E_U as the keys and values to gather information from similar users:

$$E^{uu}_{u_i} = Attn_{uu}(E_{u_i}, E_U, E_U) \tag{3.13}$$

2. **Product-to-Product Attention** We use E_{p_j} as the query and E_P as the keys and values to gather information from similar products:

$$E^{pp}_{p_j} = Attn_{pp}(E_{p_j}, E_P, E_P) \tag{3.14}$$

3. **User-to-Product Attention** We use E_{u_i} as the query and E_P as the keys and values to gather information from products associated with u_i:

$$E^{up}_{u_i} = Attn_{up}(E_{u_i}, E_P, E_P) \tag{3.15}$$

4. **Product-to-User Attention** We use E_{p_j} as the query and E_U as the keys and values to gather information from users associated with p_j:

$$E^{pu}_{p_j} = Attn_{pu}(E_{p_j}, E_U, E_U) \tag{3.16}$$

We also employ two *Multi-head Attention* between E_{u_i}/E_{p_j} (query) and H_D (key and value). The corresponding outputs are $E^D_{u_i}$ and $E^D_{p_j}$. We then combine the output of the user-product cross-context module and H_{cls} to form the final

representations. In $Attn_{uu}$ and $Attn_{pp}$, we add attention masks to prevent E_{u_i} and E_{p_j} from attending to themselves. Thus we also incorporate E_{u_i} and E_{p_j} as their *self-attentive* representations:

$$\begin{aligned} H_d = & g(E^{uu}_{u_i}, E^{pp}_{p_j}, E^{up}_{u_i}, E^{pu}_{p_j}, E^{D}_{u_i}, E^{D}_{p_j}, \\ & E_{u_i}, E_{p_j}, H_{cls}) \end{aligned} \tag{3.17}$$

H_d is fed into the classification layer to obtain the sentiment label distribution.

As with method-1, we use *Cross-Entropy* to calculate the loss between our model predictions and the gold labels.

	IMDB		Yelp-2013		Yelp-2014	
	BS	LR	BS	LR	BS	LR
BERT-base	16	6e-5	16	6e-5	16	6e-5
BERT-large	8	3e-5	8	3e-5	8	3e-5
SpanBERT-base	16	6e-5	16	6e-5	16	6e-5
SpanBERT-large	8	3e-5	8	3e-5	8	3e-5
Longformer-base	16	3e-5	16	3e-5	16	3e-5
Longformer-large	4	2e-5	4	3e-5	4	3e-5

Table 3.6: The hyperparameters used to fine-tune all models on all datasets including Learning Rate (LR) and Batch Size (BS).

	IMDB		Yelp-2013		Yelp-2014	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
Vanilla BERT-base Attention	55.4	1.129	69.1	0.617	70.7	0.610
+ Our approach	**59.7**	**1.006**	**70.7**	**0.589**	**72.4**	**0.559**
Vanilla BERT-large Attention	55.7	1.070	69.9	0.590	71.3	0.579
+ Our approach	**60.3**	**0.977**	**71.8**	**0.568**	**72.3**	**0.567**
Vanilla SpanBERT-base Attention	56.6	1.055	70.2	0.589	71.3	0.571
+ Our approach	**60.2**	**1.026**	**71.5**	**0.578**	**72.6**	**0.562**
Vanilla SpanBERT-large Attention	57.6	1.009	71.6	0.563	72.5	0.556
+ Our approach	**61.0**	**0.947**	**72.7**	**0.552**	**73.7**	**0.543**
Vanilla Longformer-base Attention	56.7	1.019	71.0	0.573	72.5	0.554
+ Our approach	**59.6**	**0.990**	**72.6**	**0.558**	**73.3**	**0.548**
Vanilla Longformer-large Attention	57.0	0.967	70.7	0.571	72.2	0.555
+ Our approach	**61.8**	**0.931**	**73.5**	**0.540**	**74.3**	**0.529**

Table 3.7: Results of our approach on various PLMs on the dev sets of IMDb, Yelp-2013 and Yelp-2014. We show the results of the baseline vanilla attention model for each PLM as well as the results of the same PLM with our proposed approach. We report the average of five runs with two metrics, Accuracy (↑) and RMSE (↓).

3.3.2 Experimental Setup

The pre-trained language models we employed in experiments are *BERT-base-uncased*, *BERT-large-uncased* [Devlin et al., 2019a], *SpanBERT-base, SpanBERT-large* [Joshi et al., 2020] and *Longformer* [Beltagy et al., 2020]. We use the implementations from Huggingface [Wolf et al., 2019]. The hyperparameters are empirically selected based on the performance on the dev set. We adopt an early stopping strategy where we stop training when the performance on dev set decreases. The maximum sequence is set to 512 for all models in order to fully utilize the textual information in documents. For evaluation, we employ two metrics *Accuracy* and *RMSE* (Root Mean Square Error), which are calculated using the scripts in [Pedregosa et al., 2011][4]. All experiments are conducted on one Nvidia GeForce RTX 3090 GPU.

We show the Learning Rate and Batch Size used to train our models on all datasets in Table 3.6.

3.3.3 Results

	IMDB		Yelp-2013		Yelp-2014	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
Pre-BERT models						
UPNN [Tang et al., 2015b]	43.5	1.602	59.6	0.784	60.8	0.764
NSC [Chen et al., 2016b]	53.3	1.281	65.0	0.692	66.7	0.654
UPDMN [Dou, 2017]	46.5	1.351	63.9	0.662	61.3	0.720
CMA [Ma et al., 2017]	54.0	1.191	66.3	0.677	67.6	0.637
HCSC [Amplayo et al., 2018]	54.2	1.213	65.7	0.660	67.6	0.639
DUPMN [Long et al., 2018]	53.9	1.279	66.2	0.667	67.6	0.639
HUAPA [Wu et al., 2018]	55.0	1.185	68.3	0.628	68.6	0.626
RRP-UPM [Yuan et al., 2019]	56.2	1.174	69.0	0.629	69.1	0.621
CHIM [Amplayo, 2019]	56.4	1.161	67.8	0.641	69.2	0.622
BERT-based models						
IUPC [Lyu et al., 2020a]	53.8	1.151	70.5	0.589	71.2	0.592
MA-BERT [Zhang et al., 2021b]	57.3	1.042	70.3	0.588	71.4	0.573
Ours	**59.0**	**1.031**	**72.1**	**0.570**	**72.6**	**0.563**

Table 3.8: Experimental Results on the test sets of IMDb, Yelp-2013 and Yelp-2014. We report the average results of of five runs of two metrics Accuracy (↑) and RMSE (↓). The best performance is in bold.

In order to validate the effectiveness of our approach, we first conduct experiments

[4]https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

with several PLMs (BERT, SpanBERT and Longformer). Results on the dev sets of IMDb, Yelp-2013 and Yelp-2014 are shown in Table 3.7. We compare our approach to a vanilla user and product attention baseline where 1) the user and product representation matrices are randomly initialized and 2) we simply employ multi-head attention between user/product and document representations without the user-product cross-context module. Our approach is able to achieve consistent improvements over the baseline with all PLMs on all three datasets. For example, our approach gives improvements over the baseline of 4.3 accuracy on IMDb, 1.6 accuracy on Yelp-2013 and 1.7 accuracy on Yelp-2014 for BERT-base. Moreover, our approach can give further improvements for large PLMs such as Longformer-large: improvements of 4.8 accuracy on IMDb, 2.8 accuracy on Yelp-2013 and 2.1 accuracy on Yelp-2014. The improvements over the baseline are statistically significant ($p < 0.01$)[5].

We compare our approach to previous approaches on the test sets of IMDb, Yelp-2013 and Yelp-2014. These include pre-BERT neural baseline models using CNN [dos Santos and Gatti, 2014, Kim, 2014] and LSTM [Yang et al., 2016] – UPNN [Tang et al., 2015b], NSC [Chen et al., 2016b], UPDMN [Dou, 2017], CMA [Ma et al., 2017], HCSC [Amplayo et al., 2018], DUPMN [Long et al., 2018], HUAPA [Wu et al., 2018], RRP-UPM [Yuan et al., 2019], CHIM [Amplayo, 2019] – and two state-of-the-art models based on BERT including our method described in Section 3.2 and MA-BERT [Zhang et al., 2021b]. We use *BERT-base* for a fair comparison with IUPC and MA-BERT, which both use *BERT-base*. The results are shown in Table 3.8. Our model obtains the best performance at both accuracy and RMSE on IMDb, Yelp-2013 and Yelp-2014. Specifically, our model achieves absolute improvements in accuracy of 1.7, 1.6 and 1.2 on IMDb, Yelp-2013 and Yelp-2013 respectively compared to previous state-of-the-art results. As for RMSE, which indicates how *close* the predicted labels are to ground-truth labels, our models outperforms earlier state-of-the-art models on RMSE by 0.011 on IMDb, 0.018 on Yelp-2013 and 0.010

[5]We use a paired t-test to determine the significance of our method's improvements over the baseline models.

on Yelp-2014.

	IMDB		Yelp-2013		Yelp-2014	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
BERT	50.8	1.187	67.2	0.639	67.8	0.629
+ User-Product Information	55.4	1.129	69.1	0.617	70.7	0.610
+ Textual Information	56.9	1.089	70.1	0.593	71.9	0.563
+ User-Product Cross-Context	59.7	1.006	70.7	0.589	72.4	0.559

Table 3.9: Results of ablation studies on the dev sets of IMDb, Yelp-2013 and Yelp-2014.

Ablation Studies Results of an ablation analysis are shown in Table 3.9. The first row results are from a BERT model without user and product information. The next three rows correspond to

1. *User-Product Information*, where we use the same method in the baseline vanilla attention model in Table 3.7 to inject user-product information
2. *Textual Information*, our proposed approach of using historical reviews to initialize user and product representations.
3. *User-Product Cross-Context*, our proposed module incorporating the associations between users and products.

The results show, firstly, that user and product information is highly useful for sentiment classification, and, secondly, that both textual information of historical reviews and user-product cross-context can improve sentiment classification. *Textual Information* gives ~1 accuracy improvement on the three datasets, while giving ~0.04 RMSE improvement on IMDb and Yelp-2014, ~0.02 RMSE improvement on Yelp-2013. *User-Product Cross-Context* achieves large improvements on IMDb of 2.8 accuracy compared to the improvements on Yelp-2013 and Yelp-2014 of 0.6 and 0.5 accuracy respectively.

Varying Number of Reviews We investigate model performance with different amounts of reviews belonging to the same user/product. We randomly sample a proportion of each user's reviews (from 10% to 100%). Then we use the sampled training data, where each user only has part of their total reviews (e.g. 10%), to

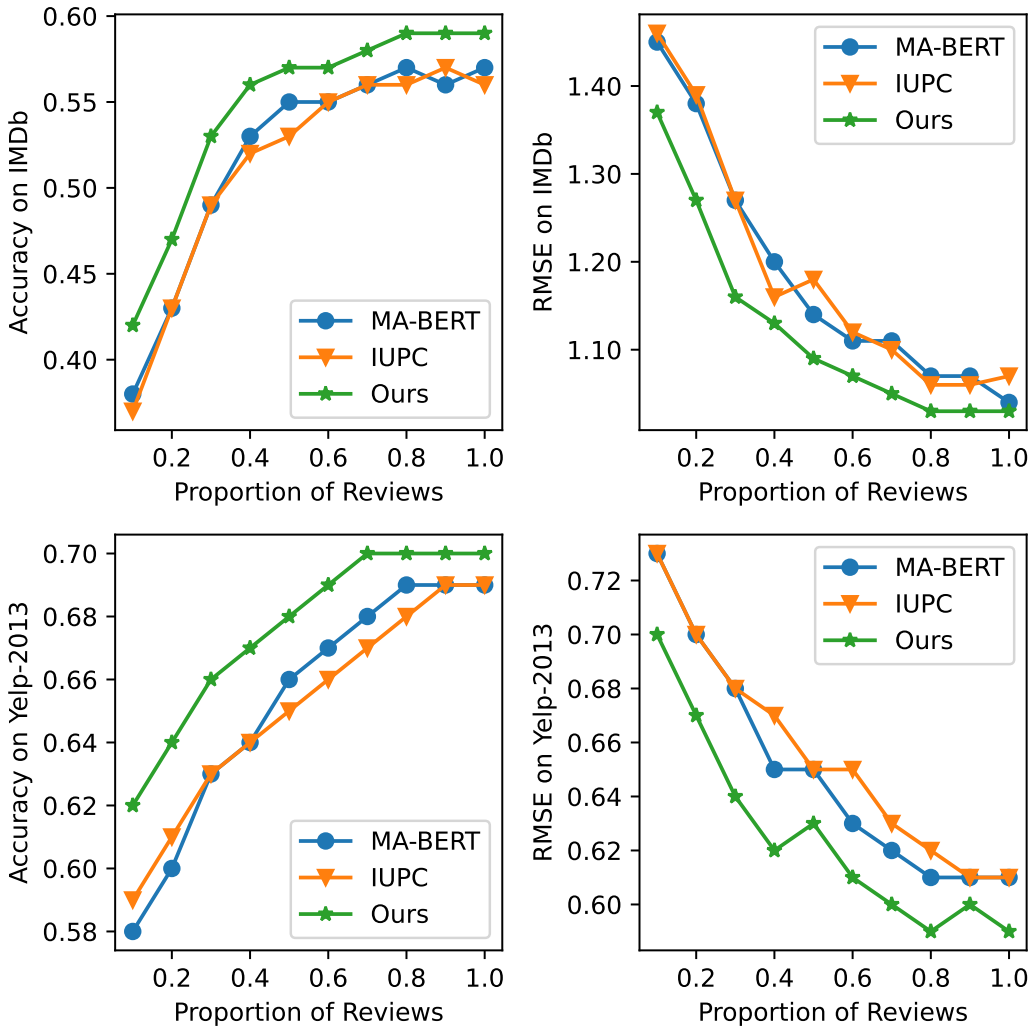

Figure 3.5: Experimental results of IUPC, MA-BERT and our approach under different proportions of reviews from 10% to 100% on the dev sets of IMDb (top) and Yelp-2013 (bottom).

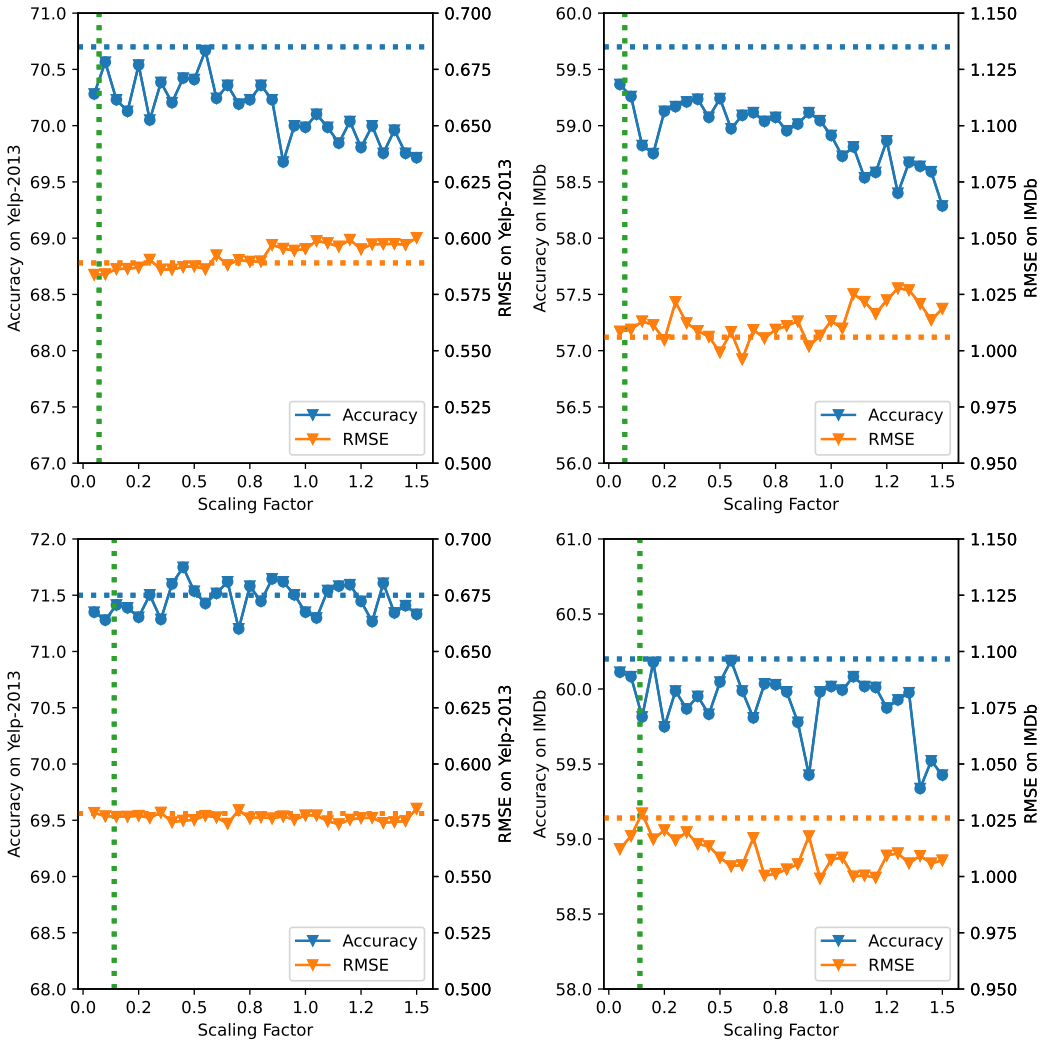

Figure 3.6: Effect of varying the scaling factor for the User and Product Matrices on the dev sets of Yelp-2013 (left) and IMDb (right). We include results of *BERT-base* (top) and *SpanBERT-base* (bottom). The left and right y-axis in each subplot represent *Accuracy* and *RMSE* respectively. The x-axis represents the scaling factor. The vertical green dashed line is the scaling factor from the Frobenius norm heuristic. The two horizontal dashed lines (blue and orange) are the accuracy and RMSE produced by the Frobenius norm heuristic respectively.

train sentiment classification models. We conduct experiments on Yelp-2013 and IMDb using IUPC (method-1 in Section 3.2), MA-BERT [Zhang et al., 2021b] and our approach. The results are shown in Figure 3.5, where the x-axis represents the proportion of reviews that we used in experiments. When the proportion of reviews lie between 10% and 50%, our approach obtains superior performance compared to MA-BERT and IUPC while the performance gain decreases when users have more reviews. The results show the advantage of our approach under a low-review scenario for users.

Scaling Factor for User/Product Matrix To investigate the effect of varying scaling factor in Equation 3.10 and 3.11 for user and product matrix. We conduct experiments with different scaling factor (see Equations 3.10 and 3.11) on the dev sets of Yelp-2013 and IMDb using *BERT-base*. We apply the same scaling factor to both user and product matrix. The results are shown in Figure 3.6, where we use scaling factor ranging from 0.05 to 1.5 with intervals of 0.05. The results show that our proposed scaling factor (green dashed lines in Figure 3.6) based on the Frobenius norm can yield competitive performance: best accuracy according to the blue dashed line. Although the RMSE of the Frobenius norm heuristic is not always the optimal, it is still a relatively lower RMSE compared to most of the other scaling factors (except the RMSE of *SpanBERT-base* on IMDb). Moreover, the Frobenius norm heuristic can reduce the efforts needed to tune the scaling factor, since the optimal scaling factor is varying for different models on different data, whereas the Frobenius norm heuristic is able to consistently provide a competitive dynamic scaling factor.

Effect of Maximum Sequence Length Document length can make document-level sentiment classification more challenging, especially for fine-grained classification, which requires model to capture the subtle expression of sentiment polarity in documents. However, PLMs often have a fixed maximum sequence length (usually 512 WordPiece [Wu et al., 2016] tokens). A commonly used method for dealing with this constraint is to only keep the first 512 tokens for documents longer than the

Review	Vanilla BERT	IUPC	MA-BERT	Ours
Took travis here for one of our first dates and just love cibo. It 's situated in a home from 1913 and has colored lights wrapped all around the trees. You can either sit inside or on the gorgeous patio. Brick oven pizza and cheese plates offered here and it 's definitely a place for a cool date. (VP)	VP (✓)	VP (✓)	VP (✓)	VP (✓)
a great sushi bar owned and operated by maggie and toshi who are both japanese. their product is always consistent and they always have a few good specials. service is great and the staff is very friendly and cheerful. value is really good particularly within their happy hour menu. our kids love it and they are always spoiled rotten by maggie and toshi so it is their favorite place. lastly we did a sake tasting there a few weeks ago and really had a great time. we all sat family style int he middle of the restaurant and got to experience some really interesting rice wines. we had a blast. great place (P)	VP (✗)	P (✓)	P (✓)	P (✓)
well , i was disappointed. i was expecting this one to be a jazzed up container store. but ... it was just average. i used to visit container store in houston near the galleria. it has a nice selection of things. people are always ready to help etc.. but , this one has an aloof sort of customer service crowd. they say nice things about your kid but do not offer to help. hmm ... i have seen similar things they were selling at ikea. the quality did seem a little better than ikea but if you are buying a laundry room shelf for your laundry detergent ... who the hell cares. its a shelf ! does n't matter if it has 15 coats of paint on the metal or 2 coats. i found one of those sistema lunch boxes that i have been looking for over here and it was on sale. will i go back ? probably not. too far out for me , plus i like ikea better (Ne)	VN (✗)	N (✗)	VN (✗)	Ne (✓)
Unfortunately tonight was the last night this location was open. The only two locations left in the valley are desert ridge and arrowhead. Please support them. (VP)	Ne (✗)	N (✗)	VN (✗)	N (✗)

Table 3.10: Example reviews from the dev sets of Yelp-2013 and the corresponding predictions of each model. Very Negative (VN), Negative (N), Neutral (Ne), Positive (P), Very Positive (VP).

maximum length. This has been shown, however, not to be the best strategy [Sun et al., 2019a], because the expression of sentiment polarity could be towards the end of a document. Therefore, in order to investigate the importance to sentiment polarity prediction of text in the tail end of a long review, we conduct experiments on the dev sets of IMDb and Yelp-2013 using Longformer-base [Beltagy et al., 2020]. We adopt various maximum sequence lengths, from 64 to up the 2048 tokens handled by

IMDb	Max Length	128	256	384	512	1024	2048
	Truncated Examples (%)	96.3	68.7	46.5	30.8	6.3	0
	Accuracy (%)	33.9	37.2	45.0	54.3	58.4	58.9
Yelp-2013	Max Length	128	256	384	512	1024	2048
	Truncated Examples (%)	63.7	29.3	13.1	5.6	0.3	0
	Accuracy (%)	63.1	66.6	68.1	68.6	69.4	69.4

Table 3.11: Results of Longformer under different maximum sequence length on the dev sets of IMDb and Yelp-2013. The truncated examples are the percentage of examples that exceed the corresponding max sequence length.

.

Longformer. In order to purely focus on review texts, we do not include user/product information in this experiment.

The results are shown in Table 3.11. When reviews longer than the maximum length are truncated, the performance of sentiment classification is substantially reduced. For example, in IMDb, when the maximum length is set to 128 and 256, 96.3% and 68.7% examples are truncated and the accuracy drops ~40% compared to the best performance. However, the effect is lower for Yelp-2013. For example, when 63.7% and 29.3% examples are truncated, the accuracy only drops ~10% and ~5% compared to the best accuracy. This is not surprising given the shorter review length of Yelp versus IMDb reviews (see Table 3.2).

Examples Some cases sampled from the dev set of Yelp-2013 and corresponding predictions from Vanilla BERT w/o user and product information, IUPC [Lyu et al., 2020a], MA-BERT [Zhang et al., 2021b] and our model are shown in Table 3.10.

1. **Example 1**This is a straightforward positive review since it clearly conveys the satisfaction towards the restaurant. Thus all models make the correct prediction.

2. **Example 2** This is similar to the first example in narrative style, but the ground-truth sentiment label is Positive rather than Very Positive since this user tends not to give very high ratings. This example shows the importance of user information.

3. **Example 3** This review conveys a very negative attitude. However, the author tends not to give very poor ratings plus the reviews this store received are not bad. With both user and product information, our model makes the correct prediction of Neutral.

4. **Example 4** All models, regardless of whether they use user and product information, predict Neutral or Negative while in fact the review label is Very Positive. This is a difficult example where the sentiment is subtly expressed.

3.4 Summary

In this chapter, we discussed how to effectively incorporate user and product context for document-level Sentiment Analysis. We propose two approaches for using the textual information in the historical reviews of users and products. Results on benchmark datasets show that our proposed approaches achieve superior performance compared to previous state–of-the-art systems, demonstrating the effectiveness of incorporating user and product context for document-level Sentiment Analysis. We have presented some answers for *RQ-1: How can we utilize the extra information in the metadata of product reviews to improve document-level sentiment analysis?*. In the next chapter, we will discuss the incorporation of linguistic and semantic knowledge for Question Generation and Question Answering. We will also present a set of experiments understanding the role of Question Answering datasets on PLMs in the next chapter.

Chapter 4

QA Experiments: Improved Unsupervised QA via improved Question Generation and Analysing QA Dataset Bias

In this chapter, we will focus on how to incorporate linguistic and semantic knowledge for unsupervised Question Answering and Question Generation as well as the effect of internal characteristics of QA datasets on QA systems' performance. We propose a novel approach using linguistic and semantic knowledge to generate questions from summarization datasets. We will discuss the effect of downstream task datasets on Pre-trained Large Language Models. We present extensive experiments investigating how internal dataset characteristics affect the performance of PLMs. This chapter is based on two papers – *Improving Unsupervised Question Answering via Summarization-Informed Question Generation* published at EMNLP 2023 Lyu et al. [2021] and *Extending the Scope of Out-of-Domain: Examining QA models in multiple subdomains* published at the *Workshop on Insights from Negative Results in NLP* at ACL 2022 [Lyu et al., 2022].

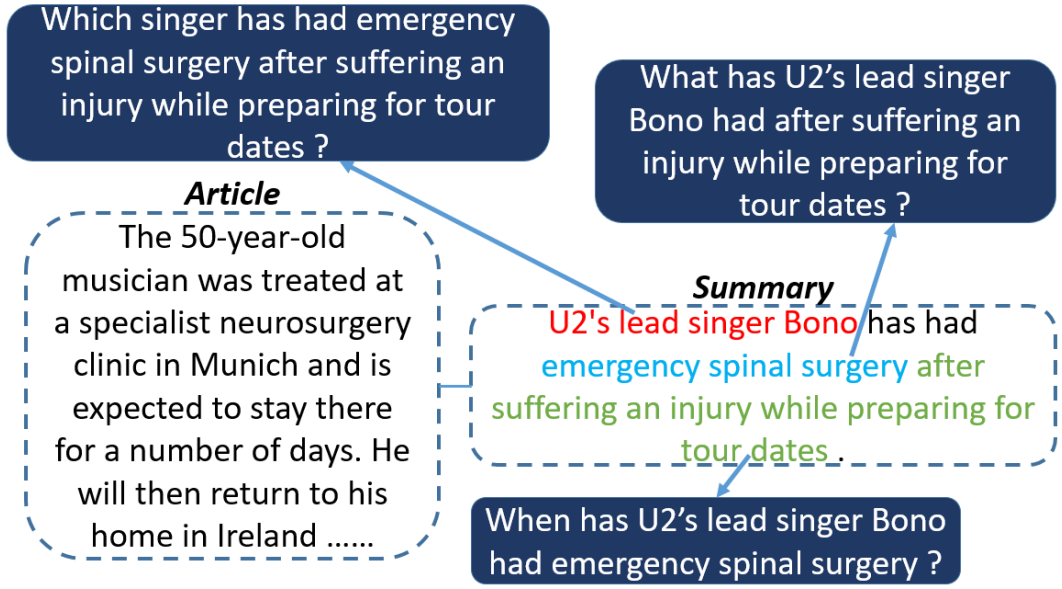

Figure 4.1: Example questions generated via heuristics informed by semantic role labeling of summary sentences using different candidate answer spans

4.1 A Novel Approach to Question Generation

The aim of Question Generation (QG) is the production of meaningful questions given a set of input passages and corresponding answers, a task with many applications including dialogue systems as well as education [Graesser et al., 2005]. Additionally, QG can be applied to Question Answering (QA) for the purpose of data augmentation [Puri et al., 2020] where labeled <*passage, answer, question*> triples are combined with synthetic <*passage, answer, question*> triples produced by a QG system to train a QA system, and unsupervised QA [Lewis et al., 2019], in which only the QG system output is used to train the QA system.

Early work on QG focused on template or rule-based approaches, employing syntactic knowledge to manipulate constituents in declarative sentences to form interrogatives [Heilman and Smith, 2009, 2010]. Although template-based methods are capable of generating linguistically correct questions, the resulting questions often lack variety and incur high lexical overlap with corresponding declarative sentences. For example, the question generated from the sentence *Stephen Hawking announced*

the party in the morning, with *Stephen Hawking* as the candidate answer span, could be *Who announced the party in the morning?*, with a high level of lexical overlap between the generated question and the declarative sentence. This is undesirable in a QA system [Hong et al., 2020] since the strong lexical clues in the question would make it a poor test of real comprehension.

Neural seq2seq models [Sutskever et al., 2014] have come to dominate QG [Du et al., 2017], and are commonly trained with <*passage, answer, question*> triples taken from human-created QA datasets [Dzendzik et al., 2021]. This limits applications to the domain and language of QA datasets. Furthermore, the process of constructing such datasets involves a significant investment of time and resources. We subsequently propose a new unsupervised approach that frames QG as a summarization-questioning process.

By employing freely available summary data, we firstly apply dependency parsing, named entity recognition and semantic role labeling to summaries, before applying a set of heuristics that generate questions based on parsed summaries. An end-to-end neural generation system is then trained employing the *original news articles* as input and the heuristically generated questions as target output.

An example is shown in Figure 4.1. The summary is used as a bridge between the questions and passages. Because the questions are generated from the summaries and not from the original passages, they have less of a lexical overlap with the passages. Crucially, however, they remain semantically close to the passages since the summaries by definition contain the most important information contained in the passages. A second advantage of this QG approach is that it does not rely on the existence of a QA dataset, and it is arguably easier to obtain summary data in a given language than equivalent QA data since summary data is created for many purposes (e.g. news, review and thesis summaries) whereas many QA datasets are created specifically for training a QA system.

In order to explore the effectiveness of our method, we train an English QG model using news summary data. We employ our QG model to generate synthetic

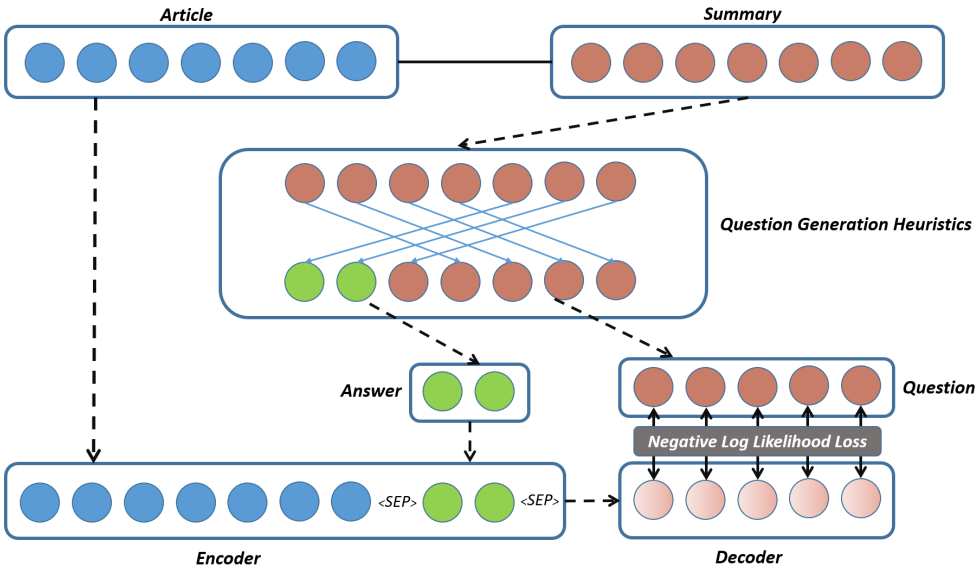

Figure 4.2: An overview of our approach where *Answer* and *Question* are generated based on *Summary* by the *Question Generation Heuristics*, the *Answer* is combined with the *Article* to form the input to the Encoder, the *Question* is employed as the ground-truth label for the outputs of the Decoder.

QA data to train a QA model in an unsupervised setting and test the approach with six English QA datasets: SQuAD1.1 [Rajpurkar et al., 2016], Natural Questions [Kwiatkowski et al., 2019], TriviaQA [Joshi et al., 2017], NewsQA [Trischler et al., 2017], BioASQ [Tsatsaronis et al., 2015] and DuoRC [Saha et al., 2018]. Experimental results show that our approach substantially improves over previous unsupervised QA models even when trained on substantially fewer synthetic QA examples.

Our contributions can be summarized as follows:

1. We propose a novel unsupervised QG approach that employs summary data and syntactic/semantic analysis, which to our best knowledge is the first work connecting text summarization and question generation in this way;
2. We employ our QG model to generate synthetic QA data achieving state-of-the-art performance even at low volumes of synthetic training data.

4.2 Methodology

Diverging from supervised neural question generation models trained on existing QA datasets, the approach we propose employs synthetic QG data, that we create from summary data using a number of heuristics, to train a QG model. We provide an overview of the proposed method in Figure 4.2. We then employ the trained QG model to generate synthetic QA data that is further employed to train an unsupervised QA model.

4.2.1 Question Generation

In order to avoid generating trivial questions that are highly similar to corresponding declarative statements, we employ summary data as a bridge connecting the generated question and the original article. The process we employ involves, firstly Dependency Parsing (DP) of summary sentences, followed by Named-Entity Recognition (NER) and finally Semantic Role Labeling (SRL). DP is firstly employed as a means of identifying the main verb (root verb), in addition to other constituents such as auxiliaries. NER is then responsible for tagging all entities in the summary sentence to facilitate discovery of the most appropriate question words to generate. The pivotal component of linguistic analysis is then SRL, employed to obtain all semantic frames for the summary sentence. Each frame consists of a verb followed by a set of arguments which correspond to phrases in the sentence. An argument could comprise, for example, an *Agent* (who initiates the action described by the verb), a *Patient* (who undergoes the action), and a set of modifier arguments such as a temporal *ARG-TMP* or locative argument *ARG-LOC*. Questions are then generated from the arguments according to argument type and NER tags, which means that wh-words can be determined jointly.

Returning to the example in Figure 4.1: given the SRL analysis *[U2's lead singer Bono* ***ARG-0****] has [had* ***VERB****] [emergency spinal surgery* ***ARG-1****] [after suffering an injury while preparing for tour dates* ***ARG-TMP****]*, the three questions shown in

Figure 4.1 can be generated based on these three arguments.

The pseudocode for our algorithm to generate questions is shown in Algorithm 1. We first obtain all dependency edges and labels (*dps*), NER tags (*ners*) and SRL

Algorithm 1: Question Generation Heuristics

```
S = summary
srl_frames = SRL(S)
ners = NER(S)
dps = DP(S)
examples = []
for frame in srl_frames do
    root_verb = dps_root
    verb = frame_verb
    if root_verb equal to verb then
        for arg in frame do
            wh* = identify_wh_word(arg, ners)
            base_verb, auxs = decomp_verb(arg, dps, root_verb)
            Q_arg = wh_move(S, wh*, base_verb, auxs)
            Q_arg = post_edit(Q_arg)
            examples.append(context, Q_arg, arg)
        end
    end
end
```

frames (*srl_frames*) of a summary sentence. We then iterate through all arguments in the frame of the *root_verb* (the verb whose dependency label is *root*) and identify appropriate wh-words (*wh**) for each argument using the function *identify_wh_word* according to its argument type and the NER tags of entities in the argument. We follow Dhole and Manning [2020] to use the standard wh-words in English associated with appropriate argument types and NER tags. We then decompose the current main verb to its base form (*base_verb*) and appropriate auxiliary words (*auxs*) in the *decomp_verb* function, before finally inserting the wh-words and the auxiliary verbs in appropriate positions using the *wh_move* function. As can be seen from Algorithm 1, a single summary sentence generates multiple questions when its SRL frame has multiple arguments.

4.2.2 Training a Question Generation Model

The summarization data we employ consists of *<passage-summary>* pairs. Questions are generated from the summaries using the heuristics described in Section 4.2.1, so that we have *<passage-summary>* pairs and *<summary-question-answer>* triples, which we then combine to form *<passage-answer-question>* triples to train a QG model. We train an end-to-end seq2seq model rather than deploying a pipeline in which the summary is first generated followed by the question to eliminate the risk of error accumulation in the generation process. By using this QG data to train a neural generation model, we expect the model to learn a combination of summarization and question generation. In other words, such knowledge can be implicitly injected into the neural generation model via our QG data.

To train the question generation model, we concatenate each passage and answer to form a sequence: *passage <SEP> answer <SEP>*, where *<SEP>* is a special token used to separate the passage and answer. This sequence is the input and the question is the target output (objective). In our experiments, we use BART [Lewis et al., 2020] described in Chapter 2.1.2.4 for generation, which is optimized by the following negative log likelihood loss function:

$$L = -\sum_{i=1}^{N} logP(q_i|C, A) \tag{4.1}$$

where q_i is the i-th token in the question, and C and A are context and answer, respectively.

4.3 Experiments

We test our idea of using summaries in question generation by applying the questions generated by our QG system in unsupervised QA. We describe the details of our experiment setup, followed by our unsupervised QA results on six English benchmark extractive QA datasets. Extractive question answering (QA) [Rajpurkar et al., 2016,

Trischler et al., 2017] is a type of QA task where the system is required to provide a concise answer to a given question by selecting a span of text, typically a sentence or a phrase, from a given document that contains the answer. Extractive QA has received significant attention from the research community due to its practical importance and its potential to solve real-world problems [Zhang et al., 2020b].

4.3.1 Question Generation

Datasets We test the proposed method using news summary data from XSUM [Narayan et al., 2018], crawled from BBC news website [1]. XSUM contains 226,711 <*passage-summary*> pairs, with each summary containing a single sentence.

QG Details We employ AllenNLP[2] [Gardner et al., 2018] to obtain dependency trees, named entities and semantic role labels for summary sentences, before further employing this knowledge to generate questions from summaries following the algorithm described in Section 4.2.1. We remove any generated <*passage-answer-question*> triples that meet one or more of the following three conditions:

1. Articles longer than 480 tokens (exceeding the maximum BART input length);
2. Articles in which fewer than 55% of tokens in the answer span are not additionally present in the passage (to ensure sufficient lexical overlap between the answer and passage);
3. Questions shorter than 5 tokens (very short questions are likely to have removed too much information)

For the dataset in question, this process resulted in a total of 14,830 <*passage-answer-question*> triples.

For training the QG model, we employ implementations of BART [Lewis et al., 2020] from Huggingface [Wolf et al., 2019]. The QG model we employ is BART-base.

[1] `www.bbc.com`
[2] https://demo.allennlp.org/

We train our system for 3 epochs with a learning rate of 3×10^{-5}, using the AdamW optimizer [Loshchilov and Hutter, 2019].

4.3.2 Unsupervised QA

Datasets We carry out experiments on six extractive QA datasets, namely, 1). SQuAD1.1 [Rajpurkar et al., 2016], a popular dataset from Stanford for machine comprehension of Wikipedia text, 2). NewsQA [Trischler et al., 2017], a dataset containing questions about news articles, 3). Natural Questions [Kwiatkowski et al., 2019], a dataset consisting of real user queries from Google, 4). TriviaQA [Joshi et al., 2017], a dataset authored by trivia enthusiasts and independently gathered evidence documents, 5). BioASQ [Tsatsaronis et al., 2015], a dataset focusing on biomedical questions, and 6). DuoRC [Saha et al., 2018], a dataset derived from movie scripts. We employ the official data of SQuAD1.1, NewsQA and TriviaQA and for Natural Questions, BioASQ and DuoRC, we employ the data released by MRQA [Fisch et al., 2019].

Unsupervised QA Training Details To generate synthetic QA training data, we make use of Wikidumps [3] by firstly removing all HTML tags and reference links, then extracting paragraphs that are longer than 500 characters, resulting in 60k paragraphs sampled from all paragraphs of Wikidumps. We employ the NER toolkits of Spacy[4] [Honnibal et al., 2020] and AllenNLP[5] [Gardner et al., 2018] to extract entity mentions in the paragraphs. We then remove <*paragraph, answer*> pairs that meet one or more of the following three conditions: 1) paragraphs with fewer than 20 or more than 480 words; 2) paragraphs with no extracted answer, or where the extracted answer is not in the paragraph due to text tokenization; 3) answers consisting of a single pronoun.

Paragraphs and answers are concatenated to form sequences of the form *passage* <*SEP*> *answer* <*SEP*>, before being fed into the trained BART-QG model to obtain

[3] https://dumps.wikimedia.org/
[4] https://spacy.io/
[5] https://demo.allennlp.org/named-entity-recognition/named-entity-recognition

corresponding questions. This results in 20k synthetic QA pairs, which are then employed to train an unsupervised QA model.

The QA model we employ is *BERT-large-whole-word-masking* (which we henceforth refer to as *BERT-large* for ease of reference). Document length and stride length are 364 and 128 respectively. The learning rate is set to 1×10^{-5}. Evaluation metrics for unsupervised QA are Exact Match (EM) that checks if the predicted answer exactly matches the ground truth answer, and F-1 score which considers both precision and recall, allowing for partial matches between the predicted and ground truth answers.

4.3.3 Results

We use the 20k generated synthetic QA pairs to train a BERT QA model and first validate its performance on the development sets of three benchmark QA datasets based on Wikipedia – SQuAD1.1, Natural Questions and TriviaQA. The results of our method are shown in Tables 4.1 and 4.2. The unsupervised baselines we compare against (also discussed in Section 2.7) are as follows:

1. Lewis et al. [2019] use unsupervised neural machine translation [Artetxe et al., 2018] to train a QG model; 4M synthetic QA examples were generated to train a QA model;
2. Li et al. [2020] employ dependency trees to generate questions and employed cited documents as passages.

For comparison, we also show the results of some supervised models fine-tuned on the corresponding training sets: Match-LSTM [Wang and Jiang], BiDAF [Seo et al., 2016], *BERT-base* and *BERT-large* [Devlin et al., 2019a].

SQuAD1.1 results are shown in Table 4.1. The results of all baseline models are taken directly from published work. As can be seen from results in Table 4.1, our proposed method outperforms all unsupervised baselines, and even exceeds the performance of one supervised model, Match-LSTM [Wang and Jiang].

	SQuAD1.1	
Models	EM	F-1
SUPERVISED MODELS		
Match-LSTM	64.1	73.9
BiDAF	66.7	77.3
BERT-base	81.2	88.5
BERT-large	84.2	91.1
UNSUPERVISED MODELS		
Lewis et al. [2019]	44.2	54.7
Li et al. [2020]	62.5	72.6
Our Method	**65.6**	**74.5**

Table 4.1: In-domain experimental results of supervised and unsupervised methods on SQuAD1.1. The highest scores of unsupervised methods are in bold.

	NQ		TriviaQA	
Models	EM	F-1	EM	F-1
SUPERVISED MODELS				
BERT-base	66.1	78.5	65.1	71.2
BERT-large	69.7	81.3	67.9	74.8
UNSUPERVISED MODELS				
Lewis et al., 2020	27.5	35.1	19.1	23.8
Li et al., 2020	31.3	48.8	27.4	38.4
Our Method	**46.0**	**53.5**	**36.7**	**43.0**

Table 4.2: In-domain experimental results: Natural Questions and TriviaQA.

Results for Natural Questions and TriviaQA are shown in Table 4.2. The results of all baseline models were produced using the released synthetic QA data to finetune a *BERT-large* model. Our method outperforms previous state-of-the-art unsupervised methods by a substantial margin, obtaining relative improvements over the best unsupervised baseline model of 47% with respect to EM and 10% F-1 on Natural Questions, and by 34% EM and 12% F-1 on TriviaQA.

In summary, our method achieves the best performance (both in terms of EM and F-1) out of three unsupervised models on all three tested datasets. Furthermore, this high performance is possible with as few as 20k training examples. This is approximately less than 10% of the training data employed in previous work [Li et al., 2020].

	NewsQA		BioASQ		DuoRC	
	EM	F-1	EM	F-1	EM	F-1
Lewis et al. [2019]	19.6	28.5	18.9	27.0	26.0	32.6
Li et al. [2020]	33.6	46.3	30.3	38.7	32.7	41.1
Our Method	**37.5**	**50.1**	**32.0**	**43.2**	**38.8**	**46.5**

Table 4.3: Out-of-domain experimental results of unsupervised methods on NewsQA, BioASQ and DuoRC. The results of two baseline models on NewsQA are taken from Li et al. [2020] and their results on BioASQ and DuoRC are from fine-tuning a BERT-large model on their synthetic data.

Transferability of Our Generated Synthetic QA Data We also validate our method's efficacy on three out-of-domain QA datasets: NewsQA, created from news articles, BioASQ, created from biomedical articles, and DuoRC, created from movie plots, for the purpose of evaluating the transferability of the Wikipedia-based synthetic data. Results in Table 4.3 show that our proposed method additionally outperforms the unsupervised baseline models on the out-of-domain datasets, achieving F1 improvements over previous state-of-the-art methods by 3.8, 4.5 and 5.4 points respectively. It is worth noting that our data adapts very well to DuoRC, created from movie plots where the narrative style is expected to require more complex reasoning. Experiment results additionally indicate that our generated synthetic data transfers well to domains distinct from that of the original summary data.

4.4 Analysis

4.4.1 Effect of Answer Extraction

In the unsupervised QA experiments, we extracted answers from Wikipedia passages before feeding them into our QG model to obtain questions. These <*passage, answer, question*> triples constitute the synthetic data employed to train the QA model. Additionally, we wish to consider what might happen if we instead employ passages and answers taken directly from the QA training data? Doing this would mean

	SQuAD1.1		NewsQA		NQ		TriviaQA	
Models	EM	F-1	EM	F-1	EM	F-1	EM	F-1
Our Method (NER-extracted answers)†	65.6	74.5	37.5	50.1	46.0	53.5	36.7	43.0
Our Method (Human-extracted answers) ‡	**68.0**	**79.5**	**40.5**	**59.3**	**57.3**	**66.7**	**54.2**	**61.1**

Table 4.4: Comparison between synthetic data generated based on Wikipedia and synthetic data generated based on corresponding training set. † are results of QA model finetuned on synthetic data generated based on NER-extracted answers, ‡ are results of QA model finetuned on synthetic data based on the answers in the training set of SQuAD1.1, NewsQA, NQ and TriviaQA.

that the QA system is no longer considered unsupervised but we carry out this experiment in order to provide insight into the degree to which there may be room for improvement in terms of our NER-based automatic answer extraction method (described in Section 4.3.2). For example, there could well be a gap between the NER-extracted answers and human-extracted answers, and in this case, the NER could extract answers, for example, that are not entirely worth asking about or indeed miss answers that are highly likely to be asked about. Results of the two additional settings are shown in Table 4.4 – answer extraction has quite a large effect on the quality of generated synthetic QA data. When we employ the answers from the training set, the performance of the QA model is improved by 5 F-1 points for SQuAD1.1, and over 10 F-1 points for Natural Questions and TriviaQA.

4.4.2 Effect of Different Heuristics

We additionally investigate the effect of a range of alternate heuristics employed in the process of constructing the QG training data described in Section 4.2.1. Recall that the QG data is employed to train a question generator which is then employed to generate synthetic QA data for unsupervised QA.

The heuristics are defined as follows:

- `Naive-QG` only employs summary sentences as passages (instead of the original articles) and generates trivial questions in which only the answer spans are replaced with the appropriate question words. For example, for the sentence *Stephen Hawking announced the party in the morning*, with *the party* as the

Heuristics	EM	F-1
Naive-QG	31.1	43.3
Summary-QG	50.9	59.4
+Main Verb	53.8	63.6
+Wh-Movement	59.5	67.7
+Decomp-Verb	64.1	73.9
+NER-Wh	**65.4**	**74.8**

Table 4.5: Experiment results of the effects to unsupervised QA performance on SQuAD1.1 of using different heuristics in constructing QG data.

answer span, the question generated by `Naive-QG` would be *Stephen Hawking announced what in the morning?* We employ the summary sentences as input and questions as target output to form the QG training data.

- `Summary-QG` makes use of the original news articles of the summaries as passages rather than summary sentences to avoid high lexical overlap between the passage and question.

  `Summary-QG` can work with the following heuristics:

  - `Main Verb`: we only generate questions based on the SRL frame of the main verb (root verb) in the dependency tree of the summary sentences, rather than using verbs in subordinate clauses;
  - `Wh-Movement`: we move the question words to the beginning of the sentence. For example, in the sentence *Stephen Hawking announced what in the morning?* we move *what* to the beginning to obtain *what Stephen Hawking announced in the morning?*;
  - `Decomp-Verb`: the main verb is decomposed to its base form and auxiliaries, e.g *what Stephen Hawking announced in the morning?* becomes *what did Stephen Hawking announce in the morning?*
  - `NER-Wh`: we employ the NER tags to get more precise question words for an answer. For example, for the answer span *NBA player Michael Jordan*, the question words would be *which NBA player* instead of *who* or *what*.

Questions	Answer	Comments
who is the frontman of swedish rock band mhiam ?	*Mattis Malinen*	✓
which sultan has been in bosnia for more than a year ?	*Sultan Mehmed II*	✓
what is a major economic driver for the state of ohio ?	*Ohio's geographic location*	✓
in what time was the first parish council elected ?	*March 1972*	✓
what do the chattanooga area will host in 2017 ?	*the Ironman Triathlon*	✓grammar error
what have sold five cars in the uk this year ?	*Surrey Motors*	missing information
when did the first military college in the us open ?	*2009*	factual error
what has been described as a " giant fish " ?	*Darwin*	mismatch

Table 4.6: Examples of generated questions with corresponding answers.✓represents correct examples.

We employ the QG data generated by these heuristics to train QG models, which leads to six BART-QG models. We then employ these six models to further generate synthetic QA data based on the same Wikipedia data and compare their performances on the SQuAD1.1 dev set. The results in Table 4.5 show that using articles as passages to avoid lexical overlap with their summary-generated questions greatly improves QA performance. `Summary-QG` outperforms `Naive-QG` by roughly 20 EM points and 16 F-1 points. The results for the other heuristics show that they continuously improve the performance, especially `Wh-Movement` and `Decomp-Verb` which make the questions in the QG data more similar to the questions in the QA dataset.

4.4.3 Effect of the Size of the Synthetic QA Data

We investigate the effects of varying the quantity of synthetic QA data. Results in Figure 4.3 show that our synthetic data allows the QA model to achieve competitive performance even with fewer than 20k examples, which suggests that our synthetic data contains sufficient QA knowledge to enable models to correctly answer a question with less synthetic data compared to previous unsupervised methods. The data-efficiency of our approach increases the feasibility of training a QA system for a target domain where there is no labeled QA data available.

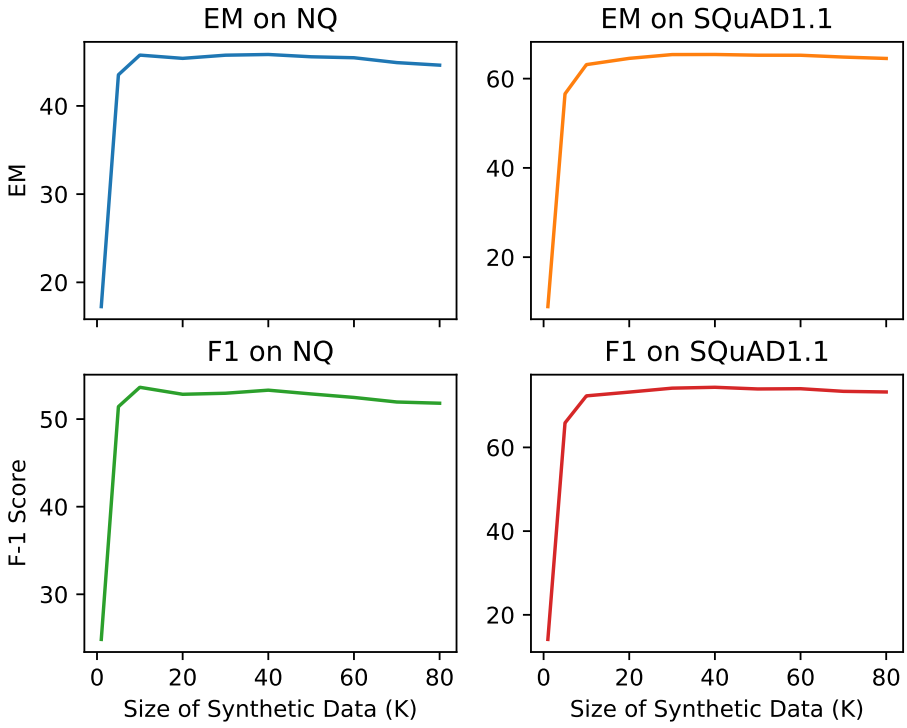

Figure 4.3: Experimental results on NQ and SQuAD1.1 of using different amount of synthetic data.

4.4.4 Few-shot Learning

We conduct experiments in a few-shot learning setting, in which we employ a limited number of labeled QA examples from the training set. We take the model trained with our synthetic QA data, the model trained with the synthetic QA data of Li et al. [2020] and a vanilla BERT model, with all QA models employing *BERT-large* [Devlin et al., 2019a]. We train these models using increasing amounts of labeled QA samples from Natural Questions (NQ) and SQuAD1.1 and assess their performance on corresponding dev sets. Results are shown in Figure 4.4 where with only a small amount of labeled data (less than 5,000 examples), our method outperforms Li et al. [2020] and *BERT-large*, clearly demonstrating the efficacy of our approach in a few-shot learning setting.

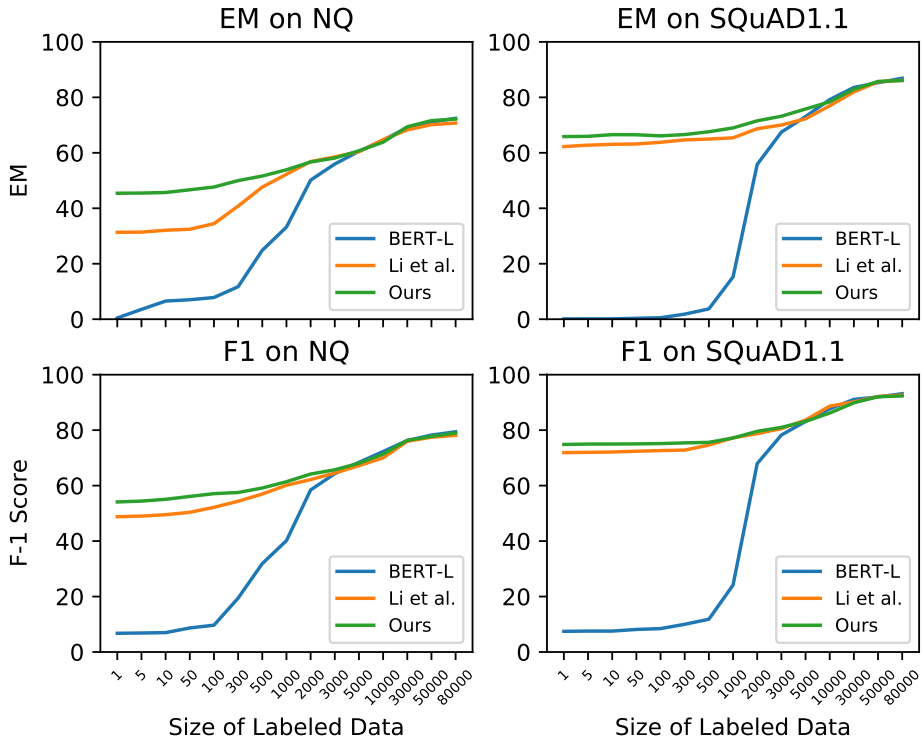

Figure 4.4: Experimental results of our method with comparison of Li et al. [2020] and *BERT-large* using different amount of labeled QA examples in the training set of NQ and SQuAD1.1.

4.4.5 Effects of Different Beam Size

We also study the effects of different beam sizes in generating synthetic questions on the performance of the downstream QA task. Experiments are conducted on the SQuAD1.1 dev set using *BERT-large*. Questions in the synthetic QA data are generated with different beam sizes using the same BART-QG model. The experimental results in Figure 4.5 show that the beam size is an important factor affecting the performance of unsupervised QA, the largest margin between the highest score (beam-15) and the lowest score (beam-1) in Figure 4.5 is close to 4 points for F-1 score.

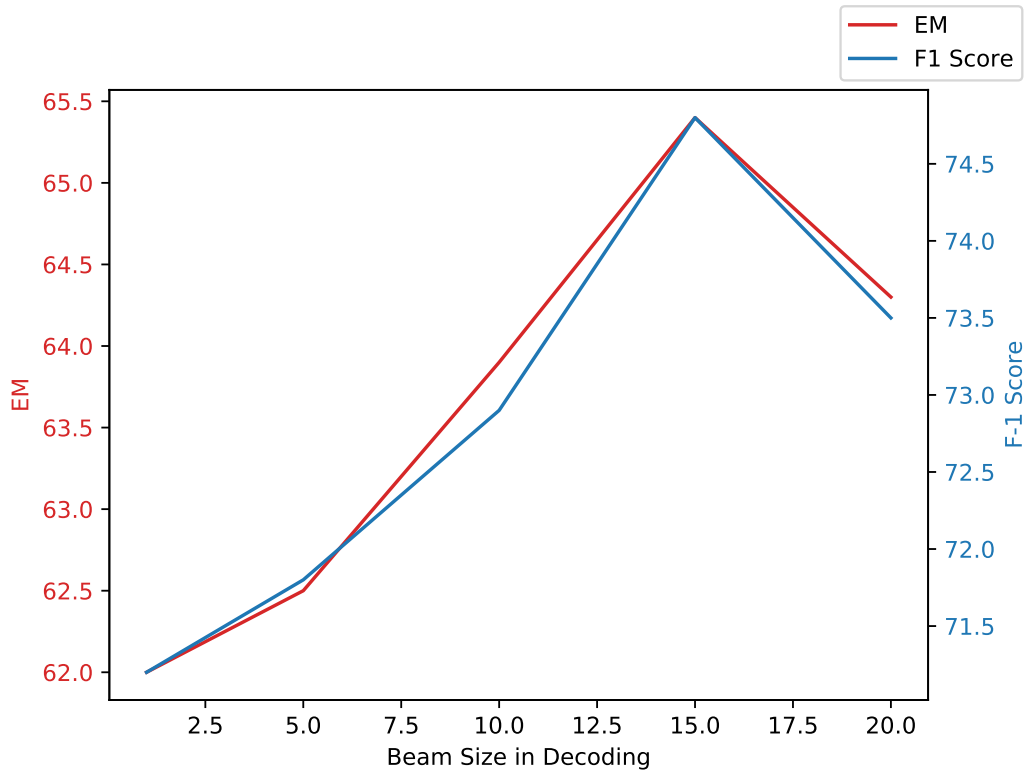

Figure 4.5: Experimental results of the effects of using different beam-size in decoding process when generating synthetic questions.

4.4.6 Question Type Distribution

We show the distribution of question types of QG data including the training set of SQuAD1.1 and our synthetic QA data in Figure 4.6, question types are defined as *What, When, Where, Who, Why, How*. The QG data has more *what, when, where* questions, indicating the existence of more SRL arguments associated with such question types in the summary sentences.

4.4.7 QG Error Analysis

Despite substantial improvements over baselines, our proposed approach inevitably still incurs error and we therefore take a closer look at the questions generated by our QG model. We manually examine 50 randomly selected questions, 31 (62%) of which were deemed high quality questions (well-structured, clear, relevant to the context

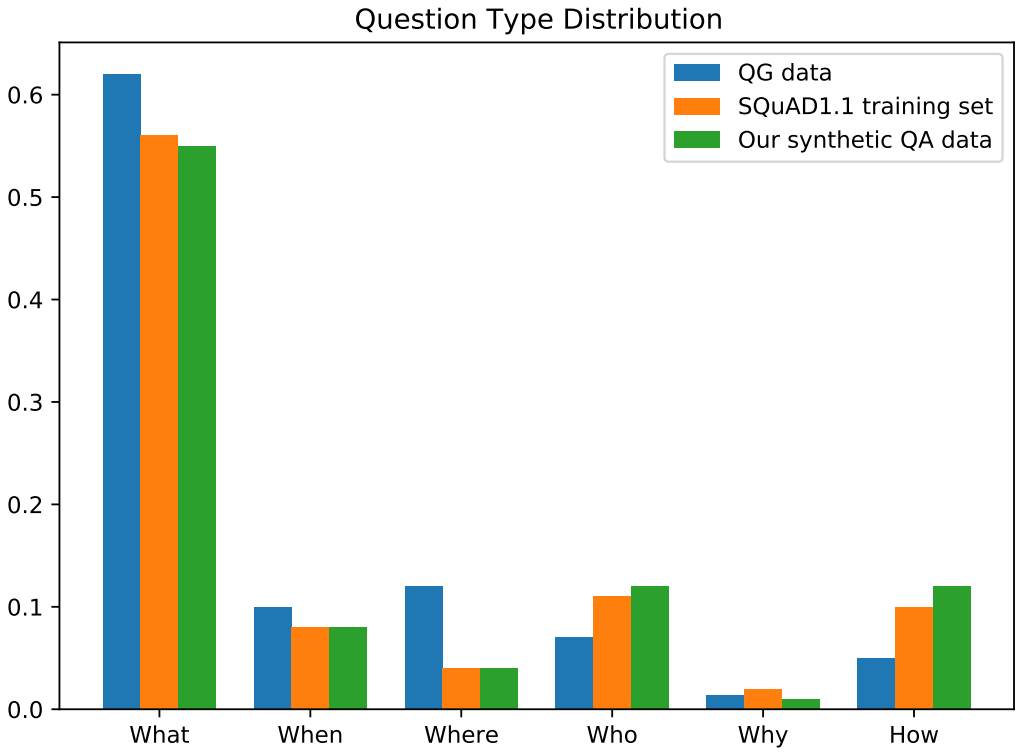

Figure 4.6: Question type distribution

and answer, and can be used as reliable QA data). The remaining 19 contain various errors with some questions containing more than one error, including mismatched wh-word and answer (6) (12%), missing information needed to locate the answer (4) (8%), factual errors (5) (10%) and grammatical errors (8) (16%) Typical examples are shown in Table 4.6.

4.5 Extending the Scope of Out-of-Domain: Examining QA models in multiple subdomains

Examining the out-of-domain performance of QA systems is an important focus of the research community due to its direct connection to the generalizability and robustness of QA systems especially in production environments [Jia and Liang, 2017, Chen et al., 2017, Talmor and Berant, 2019, Fisch et al., 2019, Shakeri et al., 2020]. Even though previous studies mostly focus on coarse-grained *general domains* [Ruder

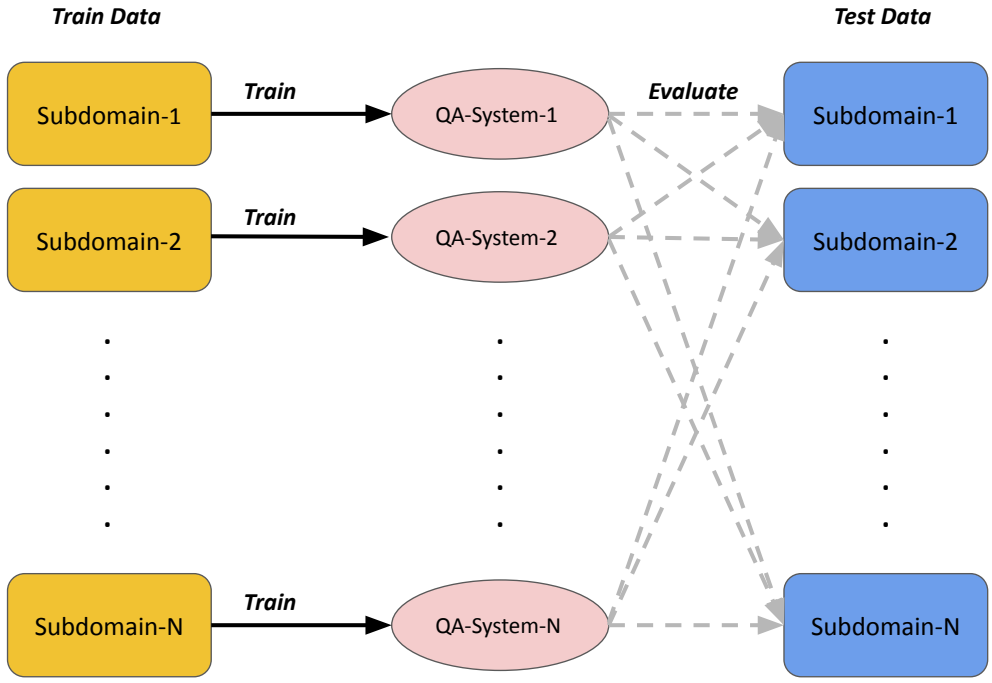

Figure 4.7: We train QA systems on each subdomain and evaluate each system on all subdomains

and Sil, 2021], the importance of finer-grained *subdomains* defined by the internal characteristics of QA datasets cannot be neglected. For example, several studies exploring specific internal characteristics of QA datasets have been carried out, including Ko et al. [2020], who reveal that the sentence-level answer position is a source of bias for QA models, and Sen and Saffari [2020] who investigate the effect of word-level question-context overlap. Building on this prior work as well as the definition and discussion of *subdomain* in Plank and Sima'an [2008], Plank [2016], Varis and Bojar [2021], we extend the scope of out-of-domain with a view to assessing the generalizability and robustness of QA systems by investigating their *out-of-subdomain* performance. As shown in Figure 4.7, we split a QA dataset into different *subdomains* based on its internal characteristics. Then we use the QA examples in each subdomain to train corresponding QA systems and evaluate their performance on all subdomains.

We focus on extractive QA as it is not only an important task in itself [Zhang et al., 2020b] but also the crucial *reader* component in the retriever-reader model

for Open-domain QA [Chen et al., 2017, Chen and Yih, 2020]. In experiments with SQuAD 1.1 [Rajpurkar et al., 2016] and NewsQA [Trischler et al., 2017], we split the data into subdomains based on *question type, text length (context, question* and *answer)* and *answer position*. We then train QA systems on each subdomain and examine their performance on each subdomain. Results show that QA systems tend to perform worse when train and test data come from different subdomains, particularly those defined by *question type, answer length* and *answer position*.

4.5.1 Experimental Setup

We employ the QA datasets, SQuAD1.1 [Rajpurkar et al., 2016] and NewsQA [Trischler et al., 2017]. For SQuAD1.1 we use the official dataset released by [Rajpurkar et al., 2016] and for NewsQA we use the data from MRQA [Fisch et al., 2019]. For question classification, we use the dataset from Li and Roth [2002]. We use the *BERT-base-uncased* model from Huggingface [Wolf et al., 2019] for both question classification and QA.

We adopt the following setup for training and evaluation: We split the original training set D into several disjoint subdomains $D_a, D_b, D_c, \ldots$; Then we sample subsets from each subdomain using sample sizes $n_1, n_2, n_3, \ldots$ in ascending order. The resulting subsets are denoted $D_a^{n_1}, D_a^{n_2}, \ldots, D_b^{n_1}, D_b^{n_2}, \ldots$. We train QA systems on each subset $D_a^{n_1}, D_a^{n_2}, \ldots$. The QA system trained on $D_a^{n_1}$ is denoted $QA_a^{n_1}$. We evaluate each QA system on the test data T which is also split into disjoint subdomains $T_a, T_b, T_c, \ldots$ similar to the training data D.

The learning rate is set to 3e-5, the maximum sequence length is set to 384 and the doc stride length is set to 128. We run the training process for 2 epochs for training each QA system. The training was conducted on one GeForce GTX 3090 GPU and the training batch size is 48.

Question type	Definition	Examples
HUM	people, individual, group, title	*What contemptible scoundrel stole the cork from my lunch ?* *Which professor sent the first wireless message in the USA ?* *Who was sentenced to death in February ?*
LOC	location, city, country, mountain, state	*Where is the Kalahari desert ?* *Where is the theology library at Notre Dame ?* *Where was Cretan when he heard screams ?*
ENTY	animal, body, color, creation, currency, disease/medical, event, food, instrument, language, plant, product, religion, sport, symbol, technique, term, vehicle	*What relative of the racoon is sometimes known as the cat-bear ?* *What is the world's oldest monographic music competition ?* *What was the name of the film about Jack Kevorkian ?*
DESC	definition, description, manner, reason	*What is Eagle 's syndrome styloid process ?* *How did Beyonce describe herself as a feminist ?* *What are suspects blamed for ?*
NUM	code, count, date, distance, money, order, other, percent, period, speed, temperature, size, weight	*How many calories are there in a Big Mac ?* *What year did Nintendo announce a new Legend of Zelda was in the works for Gamecube ?* *How many tons of cereal did Kelloggs donate ?*

Table 4.7: Definition of each question type and corresponding examples in SQuAD1.1 and NewsQA.

		LOC	ENTY	HUM	NUM	DESC
SQuAD1.1	Train set	11.4	27.6	20.7	24.5	15.5
	Dev set	10.5	27.6	21.0	23.0	17.4
NewsQA	Train set	11.4	16.9	30.0	18.8	22.6
	Dev set	12.3	16.9	32.2	17.8	20.5

Table 4.8: The percentage (%) of question types in the SQuAD1.1 and NewsQA train and dev sets.

4.5.2 Question Type

In this experiment we investigate how QA models learn from QA examples with different question types. We adopt the question classification data in Li and Roth [2002] to train a question classifier that categorizes questions into the following five classes: *HUM, LOC, ENTY, DESC, NUM* [Zhang and Lee, 2003]. The definitions and examples of each question type are shown in Table 4.7.

The training data is then partitioned into five categories according to their question type. Question type proportions for SQuAD1.1 and NewsQA are shown in Table 4.8, with a high proportion of *ENTY* and *NUM* questions in SQuAD1.1, while NewsQA has more *HUM* and *DESC* questions. We use QA examples of each question type to train a QA system, increasing the training set size in intervals of 500 from 500 to 8000. We evaluate it on the test data, which is also divided into five categories according to question type.

The F-1 scores of the QA systems trained on each question type *subdomain* are shown in Figure 4.8, for both SQuAD1.1 and NewsQA. The x-axis represents the training set size, the y-axis is the F-1 score. The results show that a QA system learns to answer a certain type of question mainly from the examples of the same question type – this is particularly true for *HUM* and *NUM* questions in SQuAD1.1 and *HUM*, *LOC* and *NUM* questions in NewsQA. Taking *NUM* questions as an example, the rightmost plots in Figure 4.8 show that performance on other question types results in only minor improvements as the training set size increases compared to the improvements on the *NUM* question type. The QA system gets most of the knowledge it needs to answer *NUM* questions from the *NUM* training examples and a similar pattern is present for other question types.

The results in Figure 4.8 show that the subdomain defined by *question type* is a source of bias when training and employing QA systems. We suspect that word use and narrative style vary over question types, injecting bias into QA systems when learning from QA examples with different question types. Therefore, we need to

improve the diversity of question types when constructing and organising QA data.

4.5.3 Text Length

The effect of text length on the performance and generalizability of neural models has been discussed in text classification and machine translation [Amplayo et al., 2019, Varis and Bojar, 2021]. As for QA, there are three components in a QA example: *context*[6]*, question, answer*. The length of each component could potentially introduce additional bias and affect how QA systems learn from QA data. For example, a short context could be *easy* since a shorter context could reduce the search space for QA models to locate the answer; on the other hand, a short context could be *hard* as it could contain less information Therefore, the following question arises naturally: are *short* and *long* contexts/questions/answers equivalent?

To answer this question, we split the employed QA datasets into *short* and *long* groups according to the median of the length of *contexts/questions/answers*.[7] Then we train disinct QA systems on the QA examples randomly sampled from *short* and *long* groups respectively, increasing the training set size in intervals of 500 from 500 to 25000.

The results are shown in Figure 4.9, where the x-axis is the training set size and the y-axis is the ratio of the performance (EM and F-1 score) of the QA_S and corresponding QA_L systems on the *text length* subdomains of *context/question/answer*. If QA_L and QA_S have no obvious difference in terms of performance on *long* and *short* groups respectively, the ratio of their performance should be close to 1.

The results show that the performance of QA_L and QA_S trained on the subdomains of *context* and *question* length have no obvious difference as all three curves converge to 1, although there are fluctuations when the sample sizes are small. In contrast, QA_L and QA_S trained on the subdomain of *answer* length behave differently – see the subplots in the two rightmost columns of Figure 4.9.

[6]We use the terms *passage* and *context* interchangeably.
[7]See the Appendix for more statistics.

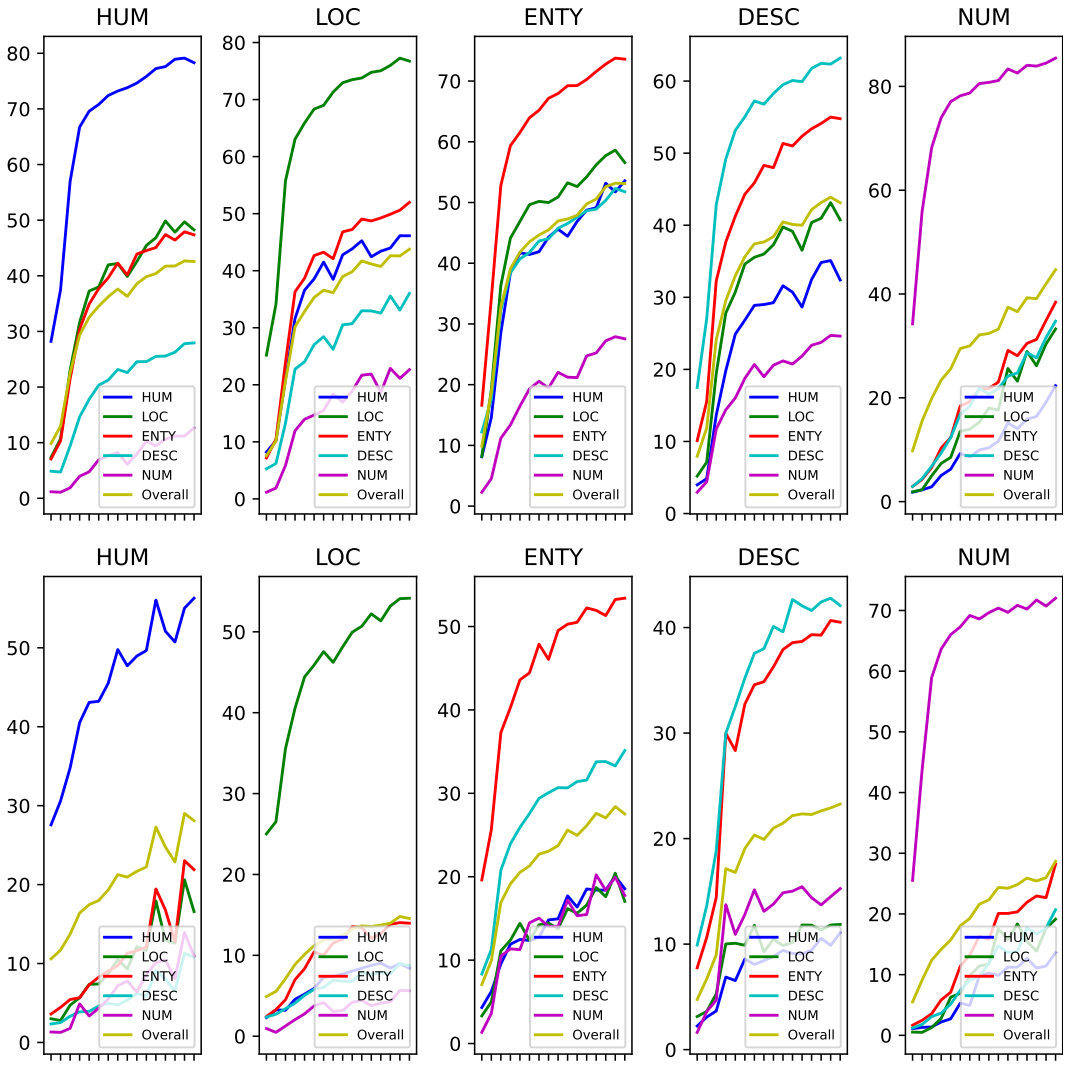

Figure 4.8: Visualization of F-1 learning curves for the QA systems trained on the *subdomains* of five question types (*HUM,LOC,ENTY,DESC,NUM*), tested on the *subdomains* for each question type and the original dev set of SQuAD1.1 (top) and NewsQA (bottom).

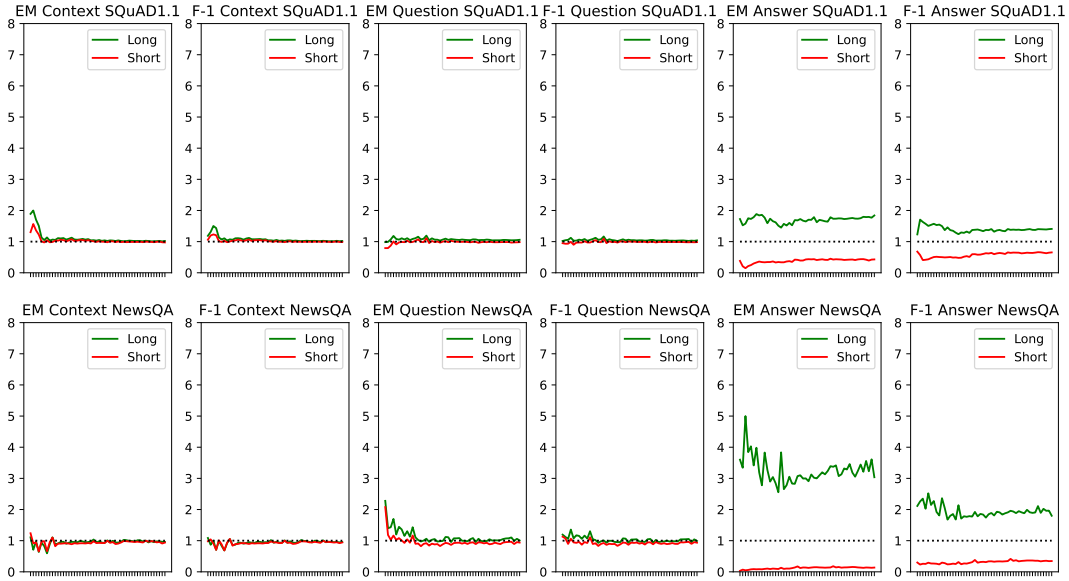

Figure 4.9: Visualization of performance (EM and F-1 score) ratio curves over *long* and *short* context, question and answer (from left to right) on SQuAD1.1 (top) and NewsQA (bottom). The *green, red* lines represent the ratio of the performance on the *long* and *short* groups. The dashed line is 1, indicating that two QA systems have the same performance. When the sample size increases, curves in *context* and *question* length converge to the dashed line, whereas there are substantial differences in the performance of QA_L and QA_S on the *answer length* subdomain.

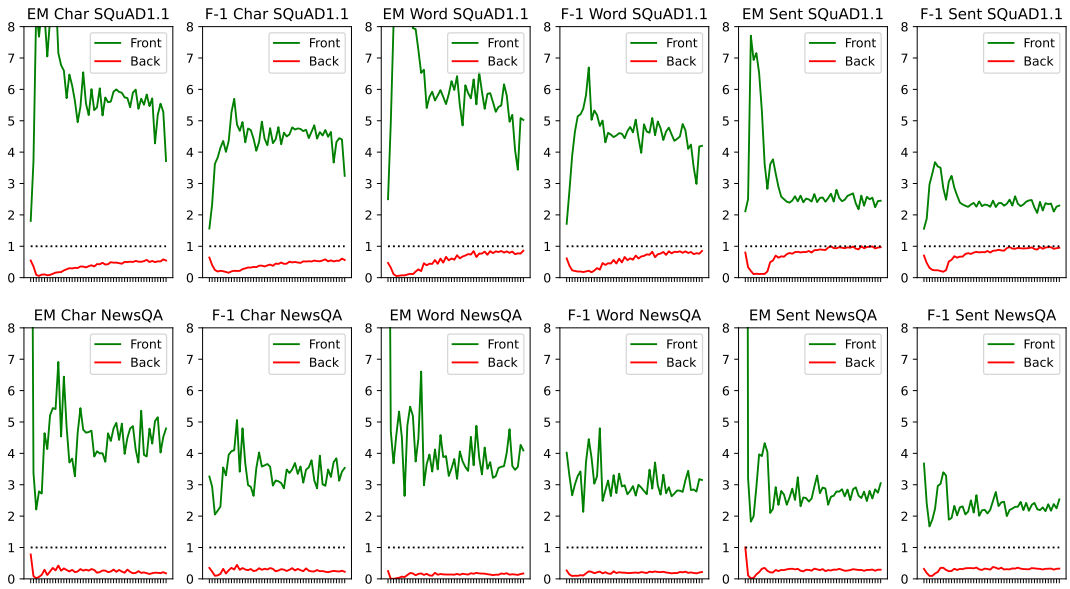

Figure 4.10: Visualization of performance (EM and F-1 score) ratio curves over *front* and *back* answer positions (char-level, word-level and sentence-level from left to right) on SQuAD1.1 (top) and NewsQA (bottom). The *green, red* lines represent the ratio of the performance on the *front* and *back* groups. The dashed line is 1, indicating that two QA systems have the same performance. The curves show that there are substantial differences in the performance of QA_F and QA_B in *answer position* subdomains, especially for character-level and word-level answer positions.

QA_L performs much better than QA_S on the test examples with *long* answers and much worse than QA_S on the test examples with *short* answers.

The results in Figure 4.9 show that the length of the answer introduces strong bias to QA systems. We think this stems from the fact that QA_L tends to predict longer answers, whereas QA_S tends to predict shorter answers, and they thus underperform in the counterpart subdomain. We show the average length of the predicted answers of QA_L and QA_S in Table 4.9. Therefore, it is important to control the length distribution of answers when constructing and organising QA datasets, especially using NER tools in the answer extraction phase since they tend to find shorter answers.

4.5.4 Answer Position

Ko et al. [2020] study the effect of sentence-level answer position. Building on their

	Context		Question		Answer	
	Long	Short	Long	Short	Long	Short
SQuAD1.1	4.03	4.13	4.00	4.23	6.41	2.78
NewsQA	5.46	5.33	5.16	5.87	9.57	3.51

Table 4.9: The average length of predicted answers of QA systems trained on *long* and *short* subdomains of *context, question* and *answer* on SQuAD1.1 and NewsQA.

analysis, we study the effect of two more types of answer position: character-level position and word-level position. We split the training set into *front* and *back* groups based on the median of the answer start positions at the character, word and sentence level.[8] Then we train QA systems on the examples sampled from the *front* ($QA_{F,char}, QA_{F,word}, QA_{F,sent}$) and *back* ($QA_{B,char}, QA_{B,word}, QA_{B,sent}$) groups respectively, increasing the training set size in intervals of 500 from 500 to 25000.

The results are shown in Figure 4.10, where the x-axis is the training set size and the y-axis is the ratio of the performance (EM and F-1 score) of QA_F and QA_B on the *answer position* subdomains at the character, word and sentence level. The results show that *answer position* is a source of bias at all three levels. QA_F performs much better than QA_B on test instances with answer positions in the *front*, whereas QA_B performs much better than QA_F on test instances with answer positions at the *back*. The effect of bias is more serious at the character and word level. We think this answer position bias is happening because words in different positions have different position embeddings, which could also affect word semantics in transformer architectures [Vaswani et al., 2017, Wang et al., 2020]. This suggests the need to make sure answer position distribution is balanced as well as the need to develop QA systems that are more robust to answer position variation.

[8]See the Appendix for more statistics.

4.6 Summary

In this chapter, we describe a novel approach for utilizing linguistic and semantic knowledge to improve Question Generation informed by summarization data for Unsupervised Question Answering. Results have shown the effectiveness of our proposed approach on various benchmark datasets for Question Answering. We also present extensive experimental results studying the effect of downstream task data on the performance of Pre-trained Large Language Models. Results demonstrate that internal characteristics of datasets pose strong bias for neural models, questioning the robustness and generalizeability of fine-tuning PLMs on downstream tasks while highlighting the importance of carefully constructing downstream datasets. This chapter partially answers *RQ-2: How can we leverage linguistic and semantic knowledge to improve Unsupervised Question Answering, and understand the role of QA data in neural model learning?*. In the next chapter, we will discuss how to incorporate information beyond text to improve multi-modal Question Answering.

Chapter 5

Semantic-aware Video Question Answering

The main focus of Chapter 5 is on how to effectively use multi-modal information from modalities beyond texts to improve Question Answering for videos. In this chapter, we will discuss two methods aiming to enhance VideoQA performance with Semantic Role Labeling (SRL).

In Chapter 4, we proposed a method for unsupervised question generation and question answering that relies on SRL. The method shows promising results. In this chapter, we extend the use of SRL even further to the video domain and explore its potential for improving VideoQA. Specifically, we will introduce two methods. The first uses SRL for improving event-level VideoQA, and is presented in Section 5.1. It was published as *Semantic-Aware Dynamic Retrospective-Prospective Reasoning for Event-Level Video Question Answering* at *ACL SRW 2023* [Lyu et al., 2023b]. The second incorporates SRL and multi-grained representations for improving VideoQA and retrieval, and is presented in Section 5.2. A paper describing this approach, entitled *Graph-Based Video-Language Learning with Multi-Grained Audio-Visual Alignment*, has been submitted to *ACM-MM 2023*.

5.1 Semantic-Aware Event-Level Video Question Answering

In general, the objective of the VQA task is to provide an answer to a visual-related question according to the content of an accompanying video. Event-level VQA (EVQA) [Xu et al., 2021b] is one specific variant of Video Question Answering (VQA) [Xu et al., 2016, Yu et al., 2018, Zhong et al., 2022b, Lyu et al., 2023a,b]. Despite significant recent progress in VQA, EVQA still remains one of the most challenging VQA-based tasks since it requires complex reasoning over the *events* across video frames [Sadhu et al., 2021, Zhong et al., 2022b, Liu et al., 2022]. To tackle the challenges in EVQA, a number of approaches have been proposed [Xu et al., 2021b, Mao et al., 2022].

Mao et al. [2022], for example, propose to construct visual scene graphs to help VQA. However, directly parsing videos to scene graphs can introduce unexpected errors due to the highly challenging cross-modal nature of the approach. Luo et al. [2022] propose a temporal-aware bidirectional attention mechanism for improving event reasoning in videos, while Zhang et al. [2022] propose a novel model named Energy-based Refined-attention Mechanism (ERM), which obtains better performance compared to previous approaches with a smaller model size. Liu et al. [2022], on the other hand, incorporate visual-linguistic causal dependencies based on Graph Convolutional Networks [Kipf and Welling, 2017] for enhancing cross-modal event reasoning for EVQA.

Despite recent advances, conventional EVQA approaches generally fail to take into account the explicit semantic connection between questions and the corresponding visual information at the event level. Therefore, we propose a new approach that takes advantage of such semantic connections, by making use of Semantic Role Labeling (SRL) [Màrquez et al., 2008, Palmer et al., 2010, He et al., 2017]. The model uses SRL information to learn an explicit semantic connection between the text-based questions and visual information in videos. Additionally, we carry out

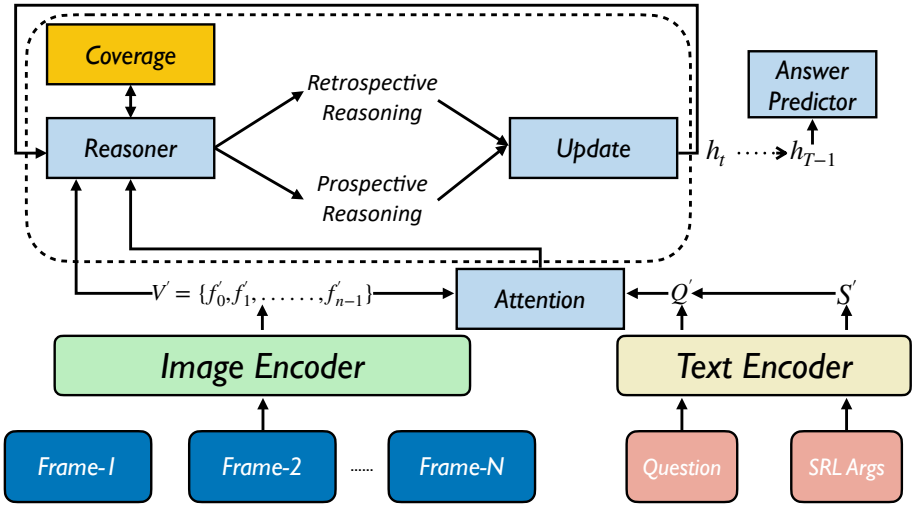

Figure 5.1: An overview of our proposed approach.

a multi-step reasoning mechanism over video frames to avoid adapting to spurious correlations and shortcuts by explicitly learning the reasoning process itself [Yi et al., 2018, Zhang et al., 2021a, Picco et al., 2021, Hamilton et al., 2022, Zhu, 2022].

Specifically, in each reasoning step, the model should explicitly decide which frame should be focused on by predicting the reasoning direction (*retrospective* or *prospective*). In terms of the question, in each reasoning step, we focus on one or more specific SRL arguments with high attention weights, and model its connection with the visual information (i.e., video frames) contained within the corresponding video. For example, for a question such as *[**ARG1**: How many cars] were [Verb: involved] [**ARG2**: in the accident?]*, the model concentrates on the ***ARG2*** when locating the accident, before determining how many cars were involved in the accident (***ARG1***). In a specific reasoning step, t, we inject the relevant visual information based on the semantic connection between the question and video frames by updating a hidden vector. This vector is ultimately expected to contain the necessary information for predicting the correct answer. In the reasoning process, we employ a *coverage mechanism* [Tu et al., 2016] to improve the coverage of the SRL arguments of the question. Namely, instead of simply focusing on a small number of specific arguments, the model is capable of including a large range of arguments.

To investigate the effectiveness of the proposed approach, we conduct experiments on a benchmark EVQA dataset: TrafficQA [Xu et al., 2021b]. Results reveal that the model achieve performance superior to that of existing baselines for a range of reasoning types (e.g., counterfactual, prospective).

5.1.1 Methodology

An overview of our approach is shown in Figure 5.1. Suppose the input of our model consists of a video V composed of n image frames sampled from it: $V = \{f_0, f_1, ..., f_{n-1}\}$, and a corresponding question $Q = \{w_0, w_1, ..., w_{m-1}\}$ with associated SRL arguments $S = \{S_0, S_1, ..., S_{N-1}\}$ where $S_i = \{w_i, w_{i+1}, ..., w_k\}$. All frames $V = \{f_0, f_1, ..., f_{n-1}\}$ are fed into an IMAGE ENCODER followed by temporal attention modeling to produce temporal-aware frame representations $V^{'} = \{f_0^{'}, f_1^{'}, ..., f_{n-1}^{'}\} \in \mathbf{R}^{n \times d}$. Meanwhile, we use a TEXT ENCODER to obtain the representations of the question with its corresponding SRL arguments: $Q^{'} \in \mathbf{R}^{1 \times d}$ and $S^{'} \in \mathbf{R}^{N \times d}$. We then perform multi-step reasoning in which we iteratively update the hidden state vector h with the visual information from frame representations based on the attention weights between them and the SRL arguments of the question. $h \in \mathbf{R}^{1 \times d}$ is updated from the initial step h_0 to the final step h_{T-1} where T is the total number of reasoning steps. Finally, we predict the most probable answer a based on h_{T-1}.

5.1.1.1 Multi-step Reasoning

Before the first reasoning step, we initialize:

$$h_0 = Attn(Q^{'}, V^{'}, V^{'}) \tag{5.1}$$

$$j = argmax(AttnWeights(Q^{'}, V^{'}, V^{'})) \tag{5.2}$$

where *Attn* serves as the q, k, v *attention*[1] modeling [Vaswani et al., 2017] and j represents the index of the frame with the highest attention weight. In each specific reasoning step t, we firstly use h_{t-1} as the *attention key* to obtain the relevant SRL argument: $S_t^{'} = Attn(h_{t-1}, S^{'}, S^{'})$. Subsequently, we infer the next focused frame by:

$$V^{focus} = Attn(r_t, V^{'}, V^{'}) \tag{5.3}$$

where $r_t = g(h_{t-1}, S_t^{'})$. Finally, we update the hidden state vector h_{t-1} based on the currently focused frame (the frame with the largest attention weight):

$$h_t = \delta(h_{t-1}, V^{focus}) \tag{5.4}$$

5.1.1.2 Retrospective-Prospective Reasoning

We propose a *Retrospective-Prospective Reasoning* mechanism for Eq.5.3 in order to explicitly decide whether the model should move to future frames (*prospective reasoning*) or move back to previous frames (*retrospective reasoning*). We obtain the *retrospective frame* V^{retro} and *prospective frame* V^{prosp} by:

$$V^{retro} = \psi(g(h_{t-1}, S_t^{'}), V^{'}, RetroMask(j)) \tag{5.5}$$

$$V^{prosp} = \phi(g(h_{t-1}, S_t^{'}), V^{'}, ProspMask(j)) \tag{5.6}$$

where ψ and ϕ are Masked Attention that are used to obtain *retrospective* and *prospective* frames, $g(h_{t-1}, S_t^{'})$ and $V^{'}$ serve as *query* and *key, value* respectively. $RetroMask(j)$ means all frames after j ($f_{i>j}$) will be masked whereas $ProspMask(j)$ means that all frames before j ($f_{i<j}$) will be masked. After obtaining V^{retro} and V^{prosp} we generate a probability:

$$p = \sigma(\lambda(V^{retro}, V^{prosp})) \tag{5.7}$$

[1] In this work, we use a low temperature τ in the *softmax* to encourage the model to assign more attention weights to the most relevant frame.

If p is larger than a pre-defined threshold α, we update $h_t = \delta(h_{t-1}, V^{retro})$,otherwise we update $h_t = \delta(h_{t-1}, V^{prosp})$ as in Eq. 5.4. The index for the next-focused frame j is also updated accordingly. The reasoning process is shown in Algorithm 2.

5.1.1.3 Coverage Mechanism

We additionally propose to employ a *coverage mechanism* [Tu et al., 2016] to encourage the model to include as many SRL arguments as possible in the reasoning process. Specifically, we track the attention distribution $C_t \in \mathbf{R}^{1\times N}$ of h_{t-1} on all SRL arguments S

$$C_t = C_{t-1} + \frac{AttnWeights([h_{t-1}; C_{t-1}], S^{'}, S^{'})}{\chi} \tag{5.8}$$

where χ represents the normalization factor.[2] We obtain the weighted $S_t^{'}$ by $S_t^{'} = Attn([h_{t-1}; C_{t-1}], S^{'}, S^{'})$ where we concatenate C_{t-1} to h_{t-1} as an additional input to the *Attn* function for the purpose of informing the model to assign more attention weights to previously less-focused SRL arguments, in order to improve the coverage for all SRL arguments.

5.1.1.4 Training Objective

For the answer prediction, we encode all answer options $A = \{a_0, ..., a_{M-1}\}$ separately and then select the one with the highest similarity with h_{T-1}. We optimize our model parameters θ using *Cross Entropy* loss:

$$J(\theta) = -\sum_i \sum_k log \frac{e^{F(a_k, h_{T-1})}}{\sum_{j=0}^{M-1} e^{F(a_j, h_{T-1})}} y_{i,k} \tag{5.9}$$

where F is the function measuring the similarity between answer candidate and h_{T-1} and we use dot product as F in experiments, and $y_{i,k}$ represents the answer label for the $i-$th example - if the correct answer for the $i-$th example is the $k-$th answer then $y_{i,k}$ is 1,otherwise it is 0.

[2] In this work, we use the number of SRL arguments of the corresponding question as the normalization factor.

Algorithm 2: Multi-step dynamic retrospective-prospective reasoning with coverage mechanism

$V^{'} = \{f_0, f_1, ..., f_{n-1}\}$: representations of video frames
$Q^{'}$: question
$S^{'}$: SRL representations of Q
T: reasoning steps
χ : normalization factor
α: threshold of the probability for using retrospective frame
$h_0 = Attn(Q^{'}, V^{'}, V^{'})$
$j = argmax(AttnWeights(Q^{'}, V^{'}, V^{'}))$
$C_0 = 0$
for i *in* T **do**
  $S_i^{'} = Attn(h_{i-1}, S^{'}, S^{'}, C_{i-1})$
  $C_i = C_{i-1} + \frac{AttnWeights(h_{i-1}, S^{'}, S^{'}, C_{i-1})}{\chi}$
  $V^{retro} = \psi(g(h_{t-1}, S_t^{'}), V^{'}, RetroMask(j))$
  $V^{prosp} = \phi(g(h_{i-1}, S_i^{'}), V^{'}, ProspMask(j))$
  $p = \sigma(f(V^{retro}, V^{prosp}))$
  if $p > \alpha$ **then**
    $h_i = \delta(h_{i-1}, V^{retro})$
    $j = argmax(\psi(g(h_{t-1}, S_t^{'}), V^{'}, RetroMask(j)))$
  else
    $h_i = \delta(h_{i-1}, V^{prosp})$
    $j = argmax(\phi(g(h_{i-1}, S_i^{'}), V^{'}, ProspMask(j)))$

5.1.2 Experimental Setup

We employ a benchmark dataset for EVQA – TrafficQA [Xu et al., 2021b] which contains 62,535 QA pairs and 10,080 videos. We follow the standard split of TrafficQA – 56,460 pairs for training and 6,075 pairs for evaluation. We further sample 5,000 examples from training data as the dev set to facilitate the selection of hyper-parameters. There are two experimental settings for TrafficQA [Xu et al., 2021b]: 1) Setting-1/2: this task is to predict whether an answer is correct for a given question based on videos; 2) Setting-1/4: this task follows the standard setup of a multiple-choice task in which the model is expected to predict the correct the answer from the four candidate options.

We use CLIP ViT-B/16 [Radford et al., 2021] [3] to initialize our image encoder and text encoder. We evenly sample 10 frames from each video in the TrafficQA dataset. The SRL parser employed in the experiments is from AllenNLP [Gardner

[3]https://openai.com/blog/clip/

Models	Setting-1/4	Setting-1/2
Q-type (random) [Xu et al., 2021b]	25.00	50.00
QE-LSTM [Xu et al., 2021b]	25.21	50.45
QA-LSTM [Xu et al., 2021b]	26.65	51.02
Avgpooling [Xu et al., 2021b]	30.45	57.50
CNN+LSTM [Xu et al., 2021b]	30.78	57.64
I3D+LSTM [Xu et al., 2021b]	33.21	54.67
VIS+LSTM [Ren et al., 2015]	29.91	54.25
BERT-VQA [Yang et al., 2020b]	33.68	63.50
TVQA [Lei et al., 2018]	35.16	63.15
HCRN [Le et al., 2020]	36.49	63.79
Eclipse [Xu et al., 2021b]	37.05	64.77
ERM [Zhang et al., 2022]	37.11	65.14
TMBC [Luo et al., 2022]	37.17	65.14
CMCIR [Liu et al., 2022]	38.58	N/A
Ours	**43.19**	**71.63**

Table 5.1: Evaluation results on TrafficQA dataset.

Method	Question Type						
	Basic	Attribution	Introspection	Counterfactual	Forecasting	Reverse	All
HCRN [Le et al., 2020]	34.17	50.29	33.40	40.73	44.58	50.09	36.26
VQAC [Kim et al., 2021]	34.02	49.43	34.44	39.74	38.55	49.73	36.00
MASN[Seo et al., 2021]	33.83	50.86	34.23	41.06	41.57	50.80	36.03
DualVGR [Wang et al., 2021a]	33.91	50.57	33.40	41.39	41.57	50.62	36.07
CMCIR [Liu et al., 2022]	36.10	52.59	38.38	46.03	48.80	52.21	38.58
Ours	**37.05**	**52.68**	**43.91**	**50.81**	**54.26**	**55.52**	**43.19**

Table 5.2: Results by various *question type* on the dev set of TrafficQA. The highest performance are in bold.

et al., 2018, Shi and Lin, 2019]. We train our model over 10 epochs with a learning rate of 1×10^{-6} and a batch size of 8. The optimizer is AdamW [Loshchilov and Hutter, 2019]. We set the maximum reasoning step T to 3 and we use a temperature τ of 0.2 in *Attention* modeling. The hyper-parameters are empirically selected based on the performance on dev set.

5.1.3 Results

The experimental results on the test set of TrafficQA are shown in Table 5.1, where we also include the previous baseline models for EVQA.[4] The results show that our

[4]Some of the baseline results are taken from [Xu et al., 2021b].

Models	Setting-1/4	Setting-1/2
Model w/o MR and CM	42.53	69.61
Model w/o CM	46.15	74.97
Model	47.38	75.83

Table 5.3: Ablation study results on TrafficQA dev set, where *MR* represents *Multi-step Reasoning* and *CM* represents *Coverage Mechanism*. MR and CM are coupled in our approach.

proposed approach obtains accuracy of 43.19 under the multiple-choice setting, which surpasses previous state-of-the-art approaches including Eclipse [Xu et al., 2021b], ERM [Zhang et al., 2022], TMBC [Luo et al., 2022] and CMCIR [Liu et al., 2022] by at least 4.5 points. Furthermore, our approach achieves an accuracy of 71.63 under Setting 1/2, outperforming previous strong baselines by at least 6 points. The results show the effectiveness of our proposed multi-step reasoning approach for event-level VideoQA.

5.1.4 Analysis

5.1.4.1 Ablation Study

We conduct experiments on the dev set of TrafficQA, investigating the contribution of both the *retrospective-prospective reasoning* and *coverage mechanism* on the performance of our proposed EVQA approach. The results are shown in Table 5.3, revealing that multi-step reasoning is critical in terms of model performance while the *coverage mechanism* can provide additional, albeit less substantial, improvements.

5.1.4.2 Results by Question Type

We take a closer look at model performance on different question types, e.g. reverse reasoning, counterfactual reasoning, etc. The results are shown in Table 5.2. They reveal that our proposed approach outperforms previous state-of-the-art models on all individual question types by a large margin with large improvements seen for *introspection*, *reverse* and *counterfactual* questions.

Reasoning Steps	Setting-1/4	Setting-1/2
Model w/ 1 step	41.57	71.46
Model w/ 2 steps	44.21	74.95
Model w/ 3 steps	47.38	75.83
Model w/ 4 steps	47.23	75.96
Model w/ 5 steps	47.15	75.87

Table 5.4: The effect of various reasoning steps.

5.1.4.3 Effect of Reasoning Steps

We study the effect of varying reasoning steps. The results are shown in Table 5.4. Increasing reasoning steps improves performance, especially from 1 step to 3 steps. Additionally, the performance (both Setting 1/4 and 1/2) is stable with reasoning steps exceeding three.

5.2 Graph-Based Video-Language Learning with Multi-Grained Audio-Visual Alignment

In the previous section, we presented a multi-step dynamic retrospective-prospective approach for Event-level VideoQA. We build on and extend this approach by using a Semantic Role Labeler and a visual scene parser to fuse text, audio and visual components in a hierarchical approach to the geenral problem of video-language learning. Video-language learning has been an active area of research in recent years, fueled by the growing availability of large-scale video datasets and advances in machine learning techniques [Ruan and Jin, 2022, Bain et al., 2021, Zhong et al., 2022b]. The task of video-language learning involves training models to understand and reason about the content of videos, including object recognition, action detection, and question answering [Zhong et al., 2022b]. Despite the progress made in this area, the integration of visual and linguistic information remains a challenging problem. While the visual information in videos can be processed using computer vision techniques, natural language processing techniques are required to interpret the

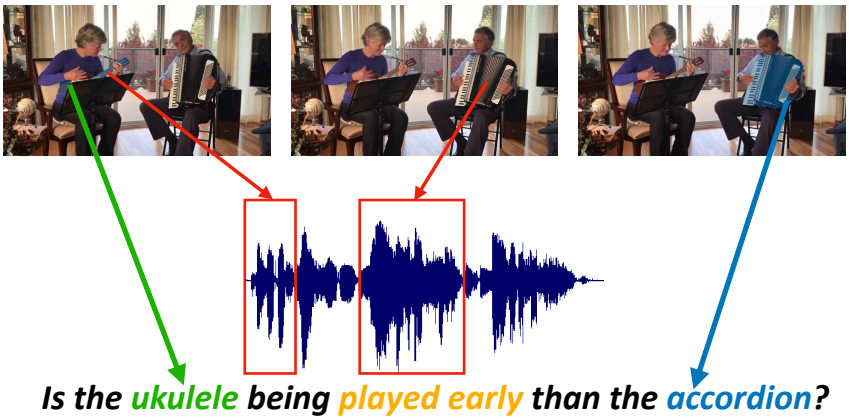

Figure 5.2: Illustration of the importance of semantic-level information and multi-grained alignment in video-language understanding. The query terms "ukulele" and "accordion" are matched to the corresponding objects in the video frames, and different segments of audio are matched to the corresponding visual information, allowing for precise determination of the order of the instrument playing.

accompanying text-based queries [Gan et al., 2022].

Graph-based approaches [Wu et al., 2020, 2023a] have emerged as a promising solution to the challenge of integrating visual and linguistic information in video-language learning. They have been successfully applied in various tasks, including image captioning, video captioning, and visual question answering. Graph-based representations are well-suited for capturing complex relationships and dependencies between objects, actions, and concepts in videos and text [Mao et al., 2022]. This is because graphs enable the representation of hierarchical and compositional structures, as well as the modeling of long-range dependencies between entities. Another key challenge in audio-aware video-language learning is the unaligned nature of audio and visual features [Alamri et al., 2019, Lee et al., 2022]. Audio features, such as speech signals or music, are typically represented in the time-frequency domain, while visual features, such as images or frames, are usually represented in the pixel or feature space. As a result, aligning audio and visual features in the same vector space with respect to the semantic-level information is challenging, especially when the features have different scales and granularities.

In this work, we propose a novel approach to video-language learning that leverages graph-based representations and multi-grained audio-visual alignment. Our approach involves transforming video and query inputs into visual-scene graphs and semantic role graphs using a visual-scene parser [Schuster et al., 2015] and a Semantic Role Labeler [Màrquez et al., 2008], as with our EVQA system described in Section 5.1. These graphs capture rich semantic-level information about the content of the video and the query. We then encode the graphs using graph neural networks to obtain enriched representations that capture the relationships between entities in the video and the query. By combining the graph-based representations of video and query, we obtain a joint representation that enhances the semantic expressivity of the inputs. Moreover, to effectively fuse the audio and visual information in videos, we propose a multi-grained alignment module that aligns the audio and visual features at multiple scales. This allows us to accurately match the relevant parts of the audio and visual features with the semantic-level information captured by the graph-based representations. An example is shown in Figure 5.2.

To evaluate the effectiveness of our proposed approach, we conduct experiments on five benchmark datasets for video retrieval and video question-answering tasks. The datasets include MSRVTT [Xu et al., 2016, Miech et al., 2018], AVSD [Alamri et al., 2019], AVQA [Yang et al., 2022], MSRVTT-MC [Yu et al., 2018] and Music-AVQA [Li et al., 2022a]. Our proposed approach achieves state-of-the-art results on all datasets, demonstrating its effectiveness. Moreover, we perform ablation studies to analyze the contribution of different components in our approach, showing that our proposed multi-grained alignment module and semantic-based content understanding significantly improve the performance of video retrieval and VideoQA.

5.2.1 Methodology

In this section, we present a more detailed description of the methodology for our proposed approach, which consists of three main steps: (1) parsing video frames and queries into semantic graphs and encoding them using Graph Neural

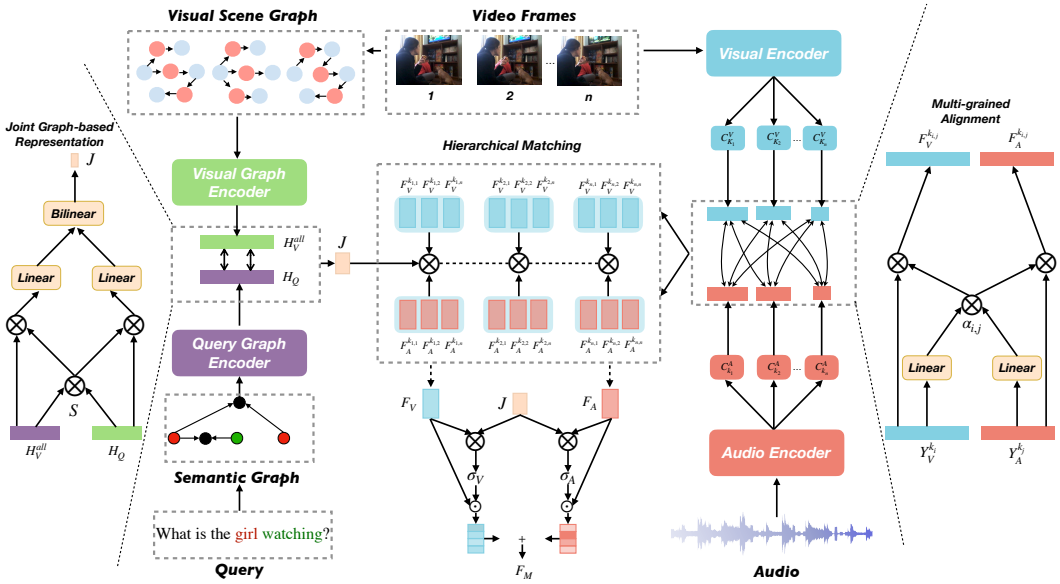

Figure 5.3: An overview of our approach for video-language learning. Our method leverages graph-based representations and multi-grained audio-visual alignment to effectively integrate visual and linguistic information. We transform video and query inputs into visual-scene graphs and semantic role graphs, encode them using graph neural networks, and combine them to obtain a video-query joint representation. Our multi-grained alignment module aligns the audio and visual features at multiple scales, allowing for accurate fusion in a way that is consistent with the semantic-level information captured by the graph-based representations.

Networks (GNNs), (2) combining these encodings to obtain a joint graph-based representation, and (3) encoding visual and audio data using Transformer encoders and fusing the resulting features through a Multi-grained Alignment (MgA) module. The final graph-based representations of the video and query are then used to match relevant visual-audio features.

5.2.1.1 Semantic Graph Parsing and Encoding

We firstly obtain the textual description of each video frame using an image captioning system, which is then parsed into a semantic graph $G_V = (V_V, E_V)$ [Schuster et al., 2015, Wu et al., 2019, Li et al., 2022b], where V_V denotes the set of nodes representing objects and actions, and E_V denotes the set of edges representing relationships between these elements.

The input query is also parsed by a Semantic Role Labeller into a semantic graph $G_Q = (V_Q, E_Q)$, representing the relationships between the query's arguments and their semantic roles.

Two separate GNNs are used to encode the video frame and query semantic graphs. These GNNs learn to map nodes and edges in the graphs to continuous feature vectors, effectively capturing the high-level semantics of the video and query. The overall general encoding process is: $H^{(k+1)} = \text{GNN}(H^{(k)}, G)$, where $H^{(k)}$ represents the graph-level features of the video and query graphs at the k-th GNN layer, and G represents the adjacency matrices of the graphs. The GNNs are applied iteratively for K layers to learn more complex graph representations. Specifically, for the node-level representations in G_V (video graph)and G_Q (query graph), we update them by:

$$h_i^{(t)} = \phi^{(t)} \left(h_i^{(t-1)}, \sum_{j \in E_i} W_i^j h_j^{(t-1)} \right) \tag{5.10}$$

where $h_i^{(t)} \in R^{1 \times w}$ is the node embeddings for node i at iteration t and w represents the dimension of node embeddings, $\phi^{(t)}$ is a node-level update function that takes as input the node embeddings for node i at iteration $t-1$, and the embeddings of the neighboring nodes in the graph $h_j^{(t-1)}$, and outputs the updated node embeddings $h_i^{(t)}$. $h_j^{(t-1)}$ is the set of neighbors of node i in the graph, E_i is the set of edges of node i, W_i^j represents the update weights for the edge E_i^j between node i and node j .

To gather the information from all the nodes in the graph, we update the graph-level representations by:

$$H^{(t)} = \psi^{(t)} \left(\sum_{i \in V} h_i^{(t)} \right) \tag{5.11}$$

where $H^{(t)} \in R^{1 \times w}$ is the graph embedding at iteration t, $\psi^{(t)}$ is a graph-level update function consisting of a linear followed by an activation function that takes as input the node embedding for all nodes in the graph at iteration t, and outputs the updated

graph embedding $H^{(t)}$. V is the set of all nodes in the graph.

Specifically, for each video frame with v node features, we obtain $\{h_1^V, ..., h_v^V\} \in R^{v \times w}$, whose graph-level features are $H_V \in R^{1 \times w}$ which is combined from $\{h_1^V, ..., h_v^V\}$ based on Equation 5.11. Then, for n video frames with n graphs $G_V^{all} = \{G_V^1, ..., G_V^n\}$, the resulting hierarchical graph-level representations are $H_V^{all} \in R^{n \times w}$, which consist of n graph-level features for each frame $\{H_V^1, ..., H_V^n\}$[5]. For query Q, we obtain its final graph-based representations $H_Q \in R^{q \times w}$ with q node features. In other words, for video frames we firstly encode each visual scene graph and then obtain their graph-level representations for each frame, resulting in n hierarchical graph-level features representing graph-based representations for n video frames. For the query semantic role graph, we directly use its node-level features to obtain fine-grained interaction between visual graphs and the query graph.

5.2.1.2 Joint Graph-based Representation

The encoded graph-based representations of the visual scene graphs of video frames and query semantic role graphs are combined to form a joint graph-based representation. This is achieved by firstly computing a similarity matrix $S \in R^{n \times q}$ between the visual graph representations and query graph representations:

$$S = H_V^{all} H_Q^T \tag{5.12}$$

Graph alignment is then performed to find the best correspondence between the nodes, resulting in a joint graph representation J:

$$J = \mathcal{E}_{Bilinear}[W_V((\frac{S}{\tau})^T H_V^{all}), W_G((\frac{S}{\tau}) H_Q)] \tag{5.13}$$

where τ represents the temperature hyperparameter for adjusting the degree of uncertainty in the similarity distribution S, $W_V \in R^{1 \times n}$ and $W_G \in R^{1 \times q}$ are weights

[5]Layer indication (k) omitted for simplicity.

matrices used to transform visual and query representations, $\mathcal{E}_{Bilinear}$ is a bilinear layer used to perform bilinear interaction between the visual graph features and query graph features, $J \in R^{1\times w}$ is the joint-graph representation.

5.2.1.3 Audio-Visual Encoding and Multi-grained Alignment

Visual and audio data from the video are separately encoded using two Transformer encoders [Radford et al., 2021]. We then perform multi-grained alignment between the visual and audio representations, resulting in feature vectors that capture the audio-visual content in multiple scales. Let X_V and X_A denote the input visual and audio features, and Y_V and Y_A denote the output features after applying the Transformer encoders:

$$Y_V = \text{Transformer}_V(X_V) \tag{5.14}$$

$$Y_A = \text{Transformer}_A(X_A) \tag{5.15}$$

where $Y_V \in R^{n\times w}$ and $Y_A \in R^{m\times w}$, n and m are the number of video frames and number of audio features respectively.

We propose a Multi-grained Alignment (MgA) module to align the visual and audio features at multiple scales. The MgA module takes the modality-specific representations of the visual and audio features as input and outputs aligned visual-audio features in multiple scales. Specifically, we employ a set of 1-D Convolutional Neural Networks (CNNs) with different kernel sizes, each with a different kernel size of k_n:

$$x^{k_n} = C_{k_n}(x)|_{n=1}^{N} \tag{5.16}$$

where x represents the input feature map, and $C_{k_n}(x)$ denotes a convolution operation with kernel size k_n, x^{k_n} is the resulting feature representations. Practically, for a specific N we employ CNNs with kernel sizes $k_1, k_2, ..., k_N$, such that $1 \leq k_1 < k_2 <$

$... < k_N \leq min(n, m)/2$. These kernel sizes are uniformly distributed within the range of 1 to $n/2$, we generally let $k_1 = 1$. The output of each CNN is a feature map with a different receptive field, which allows us to capture visual and audio features at multiple scales.

To align the visual and audio representations, we propose a cross-modal attention mechanism that operates on the output feature maps of the CNNs. Given the feature maps $Y_V^k = Y_V^{k_n}|_{n=1}^N$ and $Y_A^k = Y_A^{k_n}|_{n=1}^N$ for the visual and audio representations, respectively, we compute a set of attention maps as follows:

$$\alpha_{i,j} = \text{softmax}\left(\frac{Y_V^{k_i} W_q (Y_A^{k_j})^T}{\sqrt{w}}\right) \tag{5.17}$$

where $W_q \in R^{w \times w}$ is a learnable weight matrix, $Y_V^{k_i} \in R^{L_n^V \times w}$ and $Y_A^{k_j} \in R^{L_n^A \times w}$ are representations produced by C_{k_i} and C_{k_j}, and w is the dimensionality of the feature maps. The attention maps $\alpha_{i,j} \in R^{L_n^V \times L_n^A}$ indicate the degree of alignment between the visual and audio features at each scale k_i and k_j, where L_i^V and L_j^A are the length (the first dimension) of features produced by C_{k_i} and C_{k_j} respectively.

To obtain a fused feature representation, we compute a weighted sum of the feature maps, using the attention maps as weights:

$$F_V^{k_{i,j}} = \alpha_{i,j} Y_A^{k_j} \tag{5.18}$$

$$F_A^{k_{i,j}} = (\alpha_{i,j})^T Y_V^{k_i} \tag{5.19}$$

The fused feature representation $F_V^{k_{i,j}}$ and $F_A^{k_{i,j}}$ capture the aligned visual and audio features at multiple scales, which is expected to enhance the video representation with multi-grained visual-audio information.

5.2.1.4 Matching with Graph-Based Representations

Finally, we use the joint graph-based representation of the video and query to match the relevant parts of the aligned visual-audio features. Specifically, we perform a

hierarchical match where we firstly match and combine the multi-scale visual-audio representations within the same kernel size k_n, taking visual representations as an example:

$$\beta_{i,j}^{V} = \frac{J \cdot g(F_V^{k_{i,j}})^T}{\sum_{r=1}^{N} J \cdot g(F_V^{k_{i,r}})^T} \tag{5.20}$$

$$F_V^{k_i} = \sum_{j=1}^{N} \beta_{i,j}^{V} F_V^{k_{i,j}} \tag{5.21}$$

where $\beta_{i,j}^{V}$ represents the relevance weight between joint graph representation J and $F_V^{k_{i,j}}$ aligned between scale i and j, where g is a MeanPooling operation transforming $F_V^{k_{i,j}}$ along the first dimension to $R^{1\times w}$. Then we match the relevant parts across the features of all kernel sizes:

$$\lambda_i^V = \frac{J \cdot g(F_V^{k_i})^T}{\sum_{r=1}^{N} J \cdot g(F_V^{k_r})^T} \tag{5.22}$$

$$F_V = \sum_{i=1}^{N} \lambda_i^V g(F_V^{k_i}) \tag{5.23}$$

where λ_i^V represents the relevance weight between joint graph representation J and $F_V^{k_i}$ at scale i, and F_V is the resulting hierarchical weighted multi-grained visual representation. The same process is applied to the audio representation, resulting in two weighted multi-grained representations: F_V and F_A. Then we combine them into one multi-modal representation:

$$z_V = \sigma(J \cdot F_V^T) \tag{5.24}$$

$$z_A = \sigma(J \cdot F_A^T) \tag{5.25}$$

$$F_M = z_V F_V + z_A F_A \tag{5.26}$$

where z_V and z_A are relevance weights between J and F_V and F_A. F_M is the resulting hierarchical weighted multi-grained multi-modal representation containing the rich features aligned at multiple scales and relevant information from the graph-based representations.

5.2.1.5 Training Objectives

In video-language learning, we focus on two tasks - video retrieval and VideoQA. For video retrieval, we firstly use H_Q to interact with all video representations via GNNs and graph matching, resulting in F_M for all videos. Then we use H_Q to match with each F_M to obtain the most relevant one. For training our video retrieval systems, we perform in-batch contrastive learning [Karpukhin et al., 2020]:

$$L_{ret} = -\frac{1}{N}\sum_{i=1}^{M}\frac{e^{H_Q^i(F_M^{i,i})^T}}{\sum_{j=1}^{M} e^{H_Q^i(F_M^{i,j})^T}} \tag{5.27}$$

where $F_M^{i,j}$ is the representation generated by the interaction between the i-th query H_Q^i and the j-th video F_M^j

For VideoQA, we use Cross-Entropy loss function to train our VideoQA system in a multiple-choice setting:

$$L_{vqa} = -\frac{1}{N}\sum_{i=1}^{M}\frac{e^{H_A^i(F_M^i)^T}}{\sum_{j=1}^{T} e^{H_A^j(F_M^i)^T}} \tag{5.28}$$

where H_A^i is the answer representation and T is the size of answer candidate set.

5.2.2 Experimental Setup

5.2.2.1 Datasets

We conduct experiments on five benchmark datasets on video retrieval and VideoQA: 1) the MSRVTT dataset [Xu et al., 2016] contains 10,000 web videos with text descriptions. The dataset has two partitions: MSRVTT Original [Xu et al., 2016,

Wang et al., 2021b] and MSRVTT Miech [Miech et al., 2018]. MSRVTT Original has 6,513 clips for training, 497 clips for validation, and 2,990 clips for testing, while MSRVTT Miech has 6,656 and 1,000 clips for training and testing, respectively. We evaluated our approach on both partitions. 2) AVSD [Alamri et al., 2019].[6] In AVSD, each video is associated with a 10-round dialogue discussing the content of the corresponding video. We follow the dataset split of AVSD in [Alamri et al., 2019, Maeoki et al., 2020], 7,985 videos for training, 863 videos for validation and 1,000 videos for testing. 3) MSRVTT-MC [Xu et al., 2016, Yu et al., 2018], multi-choice VideoQA datasets - each video in MSRVTT-MC is associated with 5 candidate options. We follow the standard data split for MSRVTT-MC [Xu et al., 2016, Yu et al., 2018], where evaluation data have 2,990 videos. 4) AVQA [Yang et al., 2022] is a novel audio-visual question answering dataset focused on real-life scenario videos. It consists of 57,015 videos collected from daily audio-visual activities, alongside 57,335 specially-designed question-answer pairs that rely on clues from both modalities. The dataset contains over 158 hours of content and is divided into three subsets: 34,401 samples for the training set, 5,734 samples for the validation set, and 17,200 samples for the test set. 5) Music-AVQA [Li et al., 2022a], which is designed to assess multimodal understanding and spatio-temporal reasoning in audio-visual scenes. It includes 45,867 question-answer pairs that span 9,288 videos, amounting to more than 150 hours of content. The dataset is divided into training, validation, and testing sets containing 32,087, 4,595, and 9,185 QA pairs, respectively.

5.2.2.2 Training Setup

During training, we optimize the model parameters using AdamW [Loshchilov and Hutter, 2019], for which the ϵ is set to 1×10^{-8}. Our implementation is based on CLIP [Radford et al., 2021] from Huggingface [Wolf et al., 2019]. CLIP is used to initialize our Visual-Encoder for encoding video frames and Text-Encoder for encoding questions and answers. We employ an image captioning system from Li

[6] https://video-dialog.com

Table 5.5: Video retrieval performance results on MSRVTT-Original [Xu et al., 2016] dataset. We compare our method with state-of-the-art approaches, the results of which are taken from Lee et al. [2022]

Models	R@1	R@5	R@10	MedRank
W2VV [Dong et al., 2018]	1.1	4.7	8.1	236
Francis [Francis et al., 2019]	6.5	19.3	28.0	42
VSE++ [Faghri et al., 2017]	8.7	24.3	34.1	28
W2VV++ [Li et al., 2019c]	11.1	29.6	40.5	18
TCE [Yang et al., 2020a]	7.7	22.5	32.1	30
HGR [Chen et al., 2020b]	9.2	26.2	36.5	24
UWML [Wei et al., 2021a]	10.9	30.4	42.3	N/A
HSL [Dong et al., 2021]	11.6	30.3	41.3	17
PSM [Liu et al., 2021a]	12.0	31.7	43.0	16
T2VLAD [Wang et al., 2021b]	12.7	34.8	47.1	12
AVMA [Lee et al., 2022]	14.7	37.0	48.6	11
Our method	**19.1**	**42.7**	**55.9**	**9**

et al. [2022b][7], and the visual scene parser and Semantic Role Labeler we used in experiments are from Wu et al. [2019][8] and Gardner et al. [2018][9] respectively, the representations of nodes in semantic graphs are initialized using contextualized embeddings from BERT [Devlin et al., 2019a] and CLIP [Radford et al., 2021]. We train our system with a learning rate of 1×10^{-5} and a batch size of 16 for 20 epochs for video retrieval and a learning rate of 2×10^{-5} and a batch size of 12 for 10 epochs for VideoQA. We uniformly sample 16 frames from each video in all datasets for video retrieval and VideoQA. For efficiency consideration, we use $K = 3$ for updating GNN representations and $N = 3$ for multi-grained alignment (k_i is evenly distributed between 1 and a half of feature dimension with $k_1 = 1$) in our experiments. We use a maximum gradient norm of 5. We perform early stopping when the performance on the validation set degrades. In evaluation, we employ metrics including R@1, R@5, R@10, MedRank (Median Rank) and Mean rank for video retrieval following Lei et al. [2022], Madasu et al. [2022], for multiple-choice VideoQA we use Accuracy to measure the performance [Zhong et al., 2022b].

Table 5.6: Video retrieval performance results on MSRVTT-Miech [Miech et al., 2018] dataset. The baseline results are taken from Lee et al. [2022]

Models	R@1	R@5	R@10	MedRank
W2VV [Dong et al., 2018]	2.7	12.5	17.3	83
VSE++ [Faghri et al., 2017]	17.0	40.9	52.0	10
W2VV++ [Li et al., 2019c]	21.7	48.6	60.9	6
TCE [Yang et al., 2020a]	17.1	39.9	53.7	9
HGR [Chen et al., 2020b]	22.9	50.2	63.6	5
MMT [Gabeur et al., 2020]	20.3	49.1	63.9	6
HSL [Dong et al., 2021]	23.0	50.6	62.5	5
PSM [Liu et al., 2021a]	24.2	53.0	65.3	5
T2VLAD [Wang et al., 2021b]	26.1	54.7	68.1	4
AVMA [Lee et al., 2022]	27.8	57.3	68.7	4
Our method	**30.1**	**60.7**	**71.6**	**3**

5.2.3 Results

5.2.3.1 Video Retrieval Results

We conduct video retrieval experiments on MSRVTT-Original, MSRVTT-Miech and AVSD datasets. The results on MSRVTT-Original and MSRVTT-Miech are shown in Table 5.5 and Table 5.6. The results on AVSD are shown in Table 5.7. We compare our method with several state-of-the-art models and report the retrieval performance in terms of Recall@K (R@K), Median Rank (MedR) and Mean Rank metrics [Lei et al., 2022, Madasu et al., 2022]. Our method outperforms all the existing models in terms of all the evaluation metrics. Specifically, on the MSRVTT-Original partition, our method achieves R@1 of 19.1%, R@5 of 42.7%, R@10 of 55.9%, and MedR of 9, which are 4.4%, 5.7%, 7.3%, and 2 ranks better than the second-best performing model, AVMA [Lee et al., 2022]. Table 5.6 shows the results of our method and other state-of-the-art models on the MSRVTT-Miech partition [Miech et al., 2018]. Our method achieves state-of-the-art performance on this partition as well, with R@1=30.1%, R@5=60.7%, R@10=71.6%, and MedR=3, which are 2.3%, 3.4%, 2.9%, and 1 rank better than the second-best performing model AVMA [Lee et al., 2022]. Moreover, our method achieves significant improvements compared to the second-best

[7]https://huggingface.co/Salesforce/blip-image-captioning-base
[8]https://github.com/vacancy/SceneGraphParser
[9]https://demo.allennlp.org/semantic-role-labeling/semantic-role-labeling

Table 5.7: Experimental results on AVSD [Alamri et al., 2019] dataset. The baseline results are taken from Madasu et al. [2022], Lyu et al. [2023c]

Models	R@1	R@5	R@10	MedRank	MeanRank
LSTM [Maeoki et al., 2020]	4.2	13.5	22.1	N/A	119
FiT [Bain et al., 2021]	5.6	18.4	27.5	25	95.4
FiT (Dialogue) [Bain et al., 2021]	10.8	28.9	40.0	18	58.7
ViReD [Madasu et al., 2022]	24.9	49.0	60.8	6.0	30.3
D2V (Script) [Lyu et al., 2023c]	21.4	45.9	57.5	9.0	39.8
D2V (Summary) [Lyu et al., 2023c]	23.4	48.5	59.1	6.0	33.5
D2V (Dialogue) [Lyu et al., 2023c]	25.6	52.1	65.1	5.0	28.9
Our method	**28.9**	**59.2**	**74.5**	**4.0**	**24.2**

Table 5.8: Experimental results of VideoQA on AVQA [Yang et al., 2022] test set divided by question types. The performance of state-of-the-art approaches are taken from Yang et al. [2022].

Methods	Which	Come From	Happening	Where	Why	Before Next	When	Used For	Total Accuracy
HME [Fan et al., 2019]	82.2	85.9	79.3	76.6	57.0	80.0	57.1	76.5	81.8
HME+HAVF [Yang et al., 2022]	85.6	88.3	83.1	83.5	61.6	80.0	57.1	88.2	85.0
PSAC [Li et al., 2019b]	78.7	80.0	77.0	79.4	44.2	76.0	42.9	58.8	78.6
PSAC+HAVF [Yang et al., 2022]	89.0	91.1	83.2	81.7	61.6	82.0	52.4	76.5	87.4
LADNet [Li et al., 2019a]	81.1	87.1	76.6	81.8	67.4	78.0	47.6	76.5	81.9
LADNet+HAVF [Yang et al., 2022]	84.2	89.0	79.1	81.4	68.6	82.0	52.4	76.5	84.1
ACRTransformer [Zhang et al., 2020a]	82.5	82.8	79.4	82.5	54.7	80.0	47.6	58.8	81.7
ACRTransformer+HAVF [Yang et al., 2022]	88.5	91.7	83.9	84.9	50.0	82.0	57.1	64.7	87.8
HGA [Jiang and Han, 2020]	82.1	84.3	79.5	83.1	59.3	82.0	57.1	88.2	82.2
HGA+HAVF [Yang et al., 2022]	88.6	92.2	83.8	82.6	61.6	78.0	52.4	82.4	87.7
HCRN [Le et al., 2020]	83.7	84.1	80.2	80.9	52.3	74.0	57.1	70.6	82.5
HCRN+HAVF [Yang et al., 2022]	89.8	92.8	86.0	84.4	57.0	80.0	52.4	82.4	89.0
Our method	**93.7**	**97.3**	**90.4**	**89.5**	**61.8**	**92.0**	**64.9**	**88.2**	**93.0**

method, D2V (Dialogue) [Lyu et al., 2023c] [10]. Specifically, our method outperforms D2V (Dialogue) by 3.3% in R@1, 7.1% in R@5, and 9.4% in R@10. Additionally, our method achieves a lower MedRank and MeanRank, indicating that our method is better at retrieving relevant videos. Our method achieves a 20% lower MedRank and a 28% lower MeanRank than D2V (Dialogue). These results demonstrate that our method is effective for audio-aware video retrieval on different datasets.

5.2.3.2 VideoQA Results

We further evaluate our proposed approach on VideoQA datasets including MSRVTT-MC, AVQA and Music-AVQA. The results are shown in Table 5.9, Table 5.8 and Table 5.10. The results in Table 5.9 show that our method achieved a significant improvement in accuracy of 1.2% compared to the second-best method, HiTeA [Ye et al., 2022]. The results on AVQA in Table 5.8 demonstrate that our proposed

[10]Lyu et al. [2023c] is my own research work, which is not focusing on incorporating external knowledge into PLMs so it is not included in this thesis.

Table 5.9: Evaluation results on MSRVTT-MC [Xu et al., 2016, Yu et al., 2018] dataset.

Models	Accuracy
JSFusion [Yu et al., 2018]	83.4
ActBERT [Zhu and Yang, 2020]	85.7
ClipBERT [Lei et al., 2021]	88.2
MERLOT [Zellers et al., 2021]	90.9
VIOLET [Fu et al., 2021]	90.9
VideoCLIP [Xu et al., 2021a]	92.1
All-in-One [Wang et al., 2023]	92.0
Singularity [Lei et al., 2022]	92.1
Clover [Huang et al., 2022a]	95.2
HiTeA [Ye et al., 2022]	97.4
Our method	**98.6**

Table 5.10: Experimental results of different models on the test set of Music-AVQA[Li et al., 2022a]. We compare our proposed method with state-of-the-art approaches on Music-AVQA, of which the results are taken from Li et al. [2022a].

Method	Audio Question			Visual Question			Audio-Visual Question						All
	Counting	Comparative	Avg.	Counting	Location	Avg.	Existential	Location	Counting	Comparative	Temporal	Avg.	Avg.
FCNLSTM [Fayek and Johnson, 2020]	70.45	66.22	68.88	63.89	46.74	55.21	82.01	46.28	59.34	62.15	47.33	60.06	60.34
CONVLSTM [Fayek and Johnson, 2020]	74.07	68.89	72.15	67.47	54.56	60.94	82.91	50.81	63.03	60.27	51.58	62.24	63.65
GRU [Antol et al., 2015]	72.21	66.89	70.24	67.72	70.11	68.93	81.71	59.44	62.64	61.88	60.07	65.18	67.07
BiLSTM Attn [Zhou et al., 2016]	70.35	47.92	62.05	64.64	64.33	64.48	78.39	45.85	56.91	53.09	49.76	57.10	59.92
HCAttn [Lu et al., 2016]	70.25	54.91	64.57	64.05	66.37	65.22	79.10	49.51	59.97	55.25	56.43	60.19	62.30
MCAN [Yu et al., 2019]	77.50	55.24	69.25	71.56	70.93	71.24	80.40	54.48	64.91	57.22	47.57	61.58	65.49
PSAC [Li et al., 2019b]	75.64	66.06	72.09	68.64	69.79	69.22	77.59	55.02	63.42	61.17	59.47	63.52	66.54
HME [Fan et al., 2019]	74.76	63.56	70.61	67.97	69.46	68.76	80.30	53.18	63.19	62.69	59.83	64.05	66.45
HCRN [Le et al., 2020]	68.59	50.92	62.05	64.39	61.81	63.08	54.47	41.53	53.38	52.11	47.69	50.26	55.73
AVSD [Alamri et al., 2019]	72.41	61.90	68.52	67.39	74.19	70.83	81.61	58.79	63.89	61.52	61.41	65.49	67.44
Pano-AVQA [Yun et al., 2021]	74.36	64.56	70.73	69.39	75.65	72.56	81.21	59.33	64.91	64.22	63.23	66.64	68.93
Music-AVQA [Li et al., 2022a]	78.18	67.05	74.06	71.56	76.38	74.00	81.81	64.51	70.80	66.01	63.23	69.54	71.52
Our Method	**85.97**	**74.43**	**81.72**	**75.46**	**81.71**	**78.63**	**86.20**	**71.13**	**77.94**	**73.79**	**72.26**	**76.49**	**77.87**

method outperforms all other methods in terms of total accuracy (93.0%) and also achieves the best accuracy in all question types [Yang et al., 2022]. Furthermore, the dataset consists of three types of questions: audio, visual, and audio-visual. Each question type has several subcategories such as counting, comparative, and location.

Our proposed method outperforms all the baseline methods with an average accuracy of 77.87%. Specifically, our method achieves the highest accuracy in audio questions related to counting and comparative categories and visual questions related to location category. Additionally, our method performs significantly better than the baseline methods in audio-visual questions related to counting, comparative, and temporal categories. Our method achieves the highest accuracy in 10 out of 13 question types, including counting, comparative, existential, location, counting, temporal, and all audio-visual question types.

Table 5.11: Ablation study on MSRVTT-Original for the contributions of VG, QG and MgA modules to video retrieval task.

VG	QG	MgA	R@1	R@10	MeanRank
✗	✗	✗	13.2	43.6	50.4
✓	✗	✗	14.9	48.1	45.3
✗	✓	✗	15.4	49.2	44.9
✗	✗	✓	15.9	49.5	43.7
✓	✓	✗	16.8	52.9	39.6
✓	✗	✓	17.6	53.3	38.7
✗	✓	✓	17.1	53.2	39.1
✓	✓	✓	19.1	55.9	36.7

5.2.4 Analysis

5.2.4.1 Ablation Studies

Table 5.11 shows the ablation study results of our proposed method on the MSRVTT retrieval dataset. We experiment with three components: Visual Graph (VG), Query Graph (QG), and Multi-grained Alignment (MaG), by either including (✓) or excluding them from the model architecture (✗ means only using the vanilla representations of visual/acoustic features without the semantic graph encoding or multi-grained alignment). The evaluation metrics used are R@1, R@10, and MeanRank. The first row shows the baseline performance of the model without any of the three components. We can observe that the model achieves an R@1 score of 13.2 and a MeanRank of 50.4. By including the VG component, the model's performance improves by 1.7 points in R@1 and 4.5 points in MeanRank compared to the baseline. Similarly, by including QG, the model's performance improves by 2.2 points in R@1 and 5.7 points in MeanRank compared to the baseline. By including MaG, the model's performance improves by 2.7 points in R@1 and 6.7 points in MeanRank compared to the baseline. When we combine all three components, we achieve the best performance, with an R@1 score of 19.1, an R@10 score of 55.9, and a MeanRank score of 36.7. Compared to the second-best performing model (with VG and QG), our model improves R@1 by 2.3 points, R@10 by 2.9 points,

and MeanRank by 2.0 points. Overall, these results indicate that each component contributes to improving the performance of our model, and the combination of all three components achieves the best results on the MSRVTT retrieval dataset.

Table 5.12: Ablation study on Music-AVQA for the contributions of VG, QG and MgA modules to VideoQA task.

VG	QG	MgA	Audio	Visual	Audio-Visual
✗	✗	✗	73.58	72.97	67.81
✓	✗	✗	75.04	75.68	69.92
✗	✓	✗	76.25	75.09	70.67
✗	✗	✓	75.98	75.72	71.54
✓	✓	✗	78.85	76.91	72.73
✓	✗	✓	78.03	77.42	74.79
✗	✓	✓	81.24	77.19	75.95
✓	✓	✓	81.72	78.63	76.49

Table 5.12 shows the performance of different combinations of three components (Visual Graph, Query Graph, and Multi-grained Alignment) on the Music-AVQA dataset, evaluated on three question types, Audio, Visual, and Audio-Visual, in terms of accuracy. The first row shows the baseline performance of the model without any of the three components, achieving an Audio accuracy of 73.58, Visual accuracy of 72.97, and Audio-Visual accuracy of 67.81. Each component improves the performance of the model, with the Visual Graph component improving accuracy by 1.46 to 2.66 points, the Query Graph component improving accuracy by 2.67 to 4.66 points, and Multi-grained Alignment component improving accuracy by 2.40 to 3.68 points, depending on the modality. The combination of all three components achieves the best performance, with an Audio accuracy of 81.72, Visual accuracy of 78.63, and Audio-Visual accuracy of 76.49, outperforming the second-best performing model with VG and QG) by 0.69 to 3.69 points across different modalities. Overall, the results show that each component contributes to improving the performance of the model, and the combination of all three components achieves the best results on the Music-AVQA dataset.

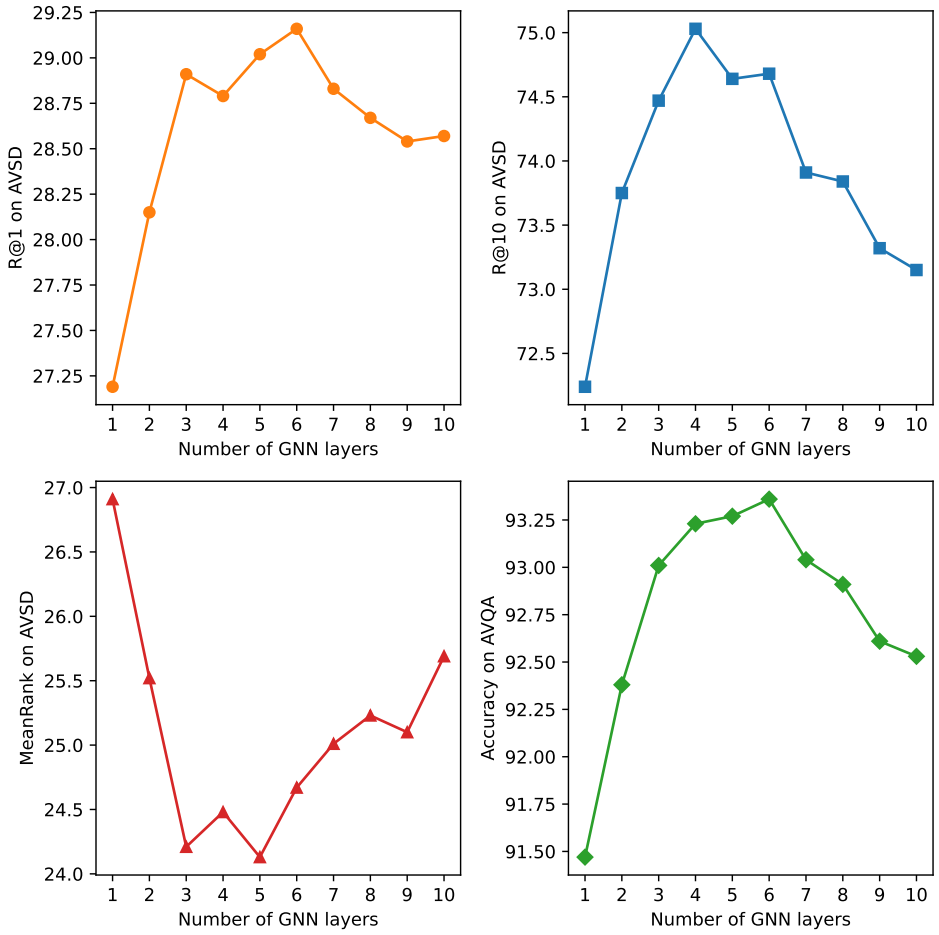

Figure 5.4: Results of the effect of the number of GNN layers on AVSD and AVQA.

5.2.4.2 Effect of the Number of GNN Layers

The integration of visual and linguistic information is a critical challenge in the field of video-language learning. To address this challenge, our proposed approach leverages graph-based representations and multi-grained audio-visual alignment. We now investigate the effect of the number of Graph Neural Network (GNN) layers on the performance of our proposed approach. We evaluate the performance of our approach on AVSD and AVQA. The motivation for conducting this experiment is to determine the optimal number of GNN layers that can effectively encode the graph-based representations of video and query inputs, and enhance the semantic expressivity of the joint representation. The results are shown in Figure 5.4. We observed that the metrics firstly improve with the number of GNN layers, then the performance plateaus before declining. We think the possible reason could be that the model with more GNN layers tends to overfit, thus resulting in a decrease in performance.

5.2.4.3 Effect of of Multi-grained Alignment Scales

The proposed Multi-grained Alignment module enables us to effectively fuse audio and visual information in a way that is consistent with the semantic-level information captured by the graph-based representations. However, the choice of scales used in the MgA module can potentially impact the performance of the model. We therefore investigate the effect of employing different scales in the multi-grained alignment module (the number of employed CNNs C_{k_n} with N different kernel sizes).

We conduct experiments on the MSRVTT-Original for video retrieval and Music-AVQA for VideoQA. The results are illustrated in Figure 5.5 consisting of four subplots, each depicting a different evaluation metric: Recall@1, Recall@10, and MeanRank for the MSRVTT-Original dataset, and Accuracy for the Music-AVQA dataset. From the results, we observe that increasing the number of scales in the MgA module contributes to improved performance across various evaluation

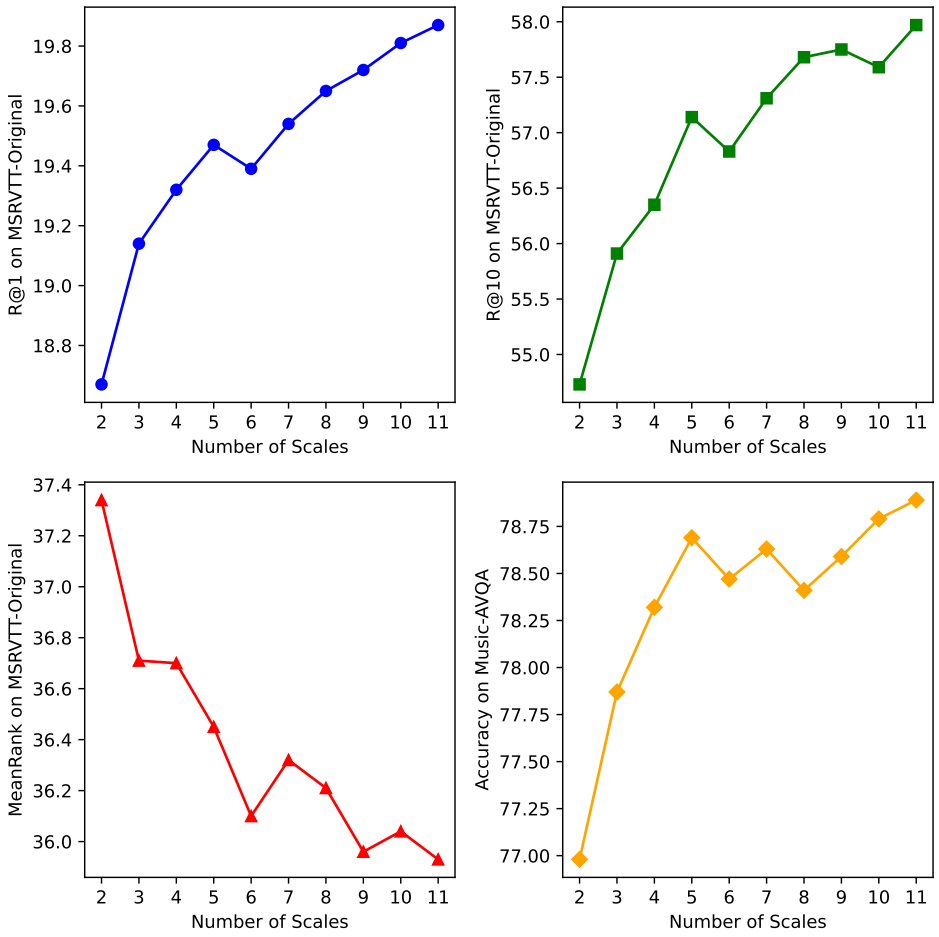

Figure 5.5: Results of the effect of employed scales of multi-grained alignment module on MSRVTT-Original and Music-AVQA.

metrics for both video retrieval and VideoQA tasks. This suggests that the MgA module effectively captures different levels of granularity and enriches the video representations produced by the model.

5.3 Summary

In this chapter, we introduced two sets of experiments on incorporating multi-modal information into Video Question Answering systems for better performance. Results on benchmark datasets have demonstrated the superior performance of our proposed approach. This chapter has provided some answers to *RQ3: How can the utilization of multi-modal information improve Video Question Answering tasks for Pre-trained Vision-Language Models?*. In the next and final chapter, we will summarise the content of the thesis and present potential directions for future work.

Chapter 6

Conclusion

6.1 Thesis Overview

This thesis explores the learning of knowledge for Pre-trained Large Language Models, focusing particularly on Sentiment Analysis and Question Answering tasks. Specifically, the research questions we proposed in this thesis mainly concentrate on how we can incorporate external knowledge to improve model performance and how we can understand the impact of fine-tuning data on model performance.

Chapter 1 introduced the research questions, provided background information, and outlined the structure of the thesis.

Chapter 2 reviewed related work on Pre-trained Large Language Models, knowledge-enhanced PLMs, Document-level Sentiment Analysis with user and product context and Question Answering, including Multi-modal Question Answering.

Chapter 3 presented our proposed approaches for incorporating textual information from historical reviews of users and products to improve Document-level Sentiment Analysis.

Chapter 4 presented our approach for utilizing linguistic and semantic knowledge to improve Unsupervised Question Answering via summarization-informed Question Generation. We also present analysis studying the effect of downstream datasets on the performance of PLMs on Question Answering tasks.

Chapter 5 described our approaches for effectively incorporating multi-modal information to improve Video Question Answering performance.

6.2 Answering Our Research Questions

- **Research Question 1 (RQ1).** *How can we utilize the extra information in the metadata of product reviews to improve document-level sentiment analysis?* To address this question, we proposed novel and effective methods for explicitly incorporating textual information from a user's historical reviews and product-specific reviews to enhance the performance of Pre-trained Language Models (PLMs) in fine-grained Document-Level Sentiment Analysis for English.

- **Research Question 2 (RQ2).** *How can we leverage linguistic and semantic knowledge to improve Unsupervised Question Answering, and understand the role of QA data in neural model learning?* In response to this question, we introduced a novel approach that utilizes summarization datasets combined with linguistic and semantic knowledge to construct synthetic Question Generation (QG) datasets. These datasets are then used to train a QG system, which generates synthetic QA examples for Unsupervised QA tasks. Furthermore, we presented a set of experiments investigating the effect of internal characteristics of QA datasets on model performance, demonstrating the strong bias introduced by question type, answer length and answer position.

- **Research Question 3 (RQ3).** *How can the utilization of multi-modal information improve Video Question Answering tasks for Pre-trained Vision-Language Models?* To tackle this question, we developed techniques for effectively integrating multi-modal information – such as incorporating Semantic Role Labeling (SRL) knowledge for injecting semantic information into question representations and multi-grained alignment for encoding visual and audio features – into pre-trained vision-language models, thereby enhancing their performance on Video Question Answering tasks.

These contributions collectively demonstrate the potential of incorporating external knowledge, such as textual metadata, linguistic and semantic knowledge,

and multi-modal information, into PLMs. By minimizing model architecture modifications, we have shown that it is possible to enhance the performance and capabilities of PLMs, providing valuable insights for various NLP tasks. Overall, our research findings address the main research question, *advancing the understanding and practical application of incorporating knowledge beyond text into PLMs*, by showcasing novel approaches for incorporating knowledge beyond text into PLMs, improving their performance in Sentiment Analysis, Unsupervised Question Answering, and Video Question Answering tasks.

6.3 Contributions

The research presented in this PhD thesis has resulted in significant contributions to the fields of Sentiment Analysis, Unsupervised Question Answering, and Video Question Answering. The main contributions of each chapter are summarized below:

- In Chapter 3, we proposed approaches for explicitly incorporating textual user and product context from historical reviews to improve Sentiment Analysis. A set of experiments presented in Chapter 3 demonstrate that our approaches achieve superior performance compared to state-of-the-art systems. The proposed approaches have resulted in two papers accepted to COLING 2020 and ACL 2023.

- Chapter 4 presented a novel method for improving Unsupervised Question Answering using summarization-informed Guestion Generation. Empirical results show that our method obtains state-of-the-art performance on benchmark datasets. We also present a set of experiments investigating the effect of internal characteristics of QA datasets on model performance, with results suggesting that internal characteristics of QA datasets could introduce strong bias to QA systems. The experiments conducted in Chapter 4 resulted in two papers accepted to EMNLP 2021 and the Insights Workshop at ACL 2022.

- Chapter 5 proposed two approaches: 1) incorporating Semantic Role Labeling knowledge to improve the reasoning of VideoQA systems. 2) a graph-based video-language learning approach with multi-grained audio-visual alignment. Experiments conducted on benchmark datasets demonstrate that our approaches substantially improved the performance of VideoQA systems. The research presented in Chapter 5 resulted in two papers accepted to ACL SRW 2023 and ACM-MM 2023.

6.4 Limitations

In this work, we have endeavored to highlight the limitations and challenges associated with the current state of Pre-trained Large Language Models, especially in the context of Sentiment Analysis and Question Answering tasks. Some specific limitations include:

- The difficulty in incorporating external knowledge effectively, such as linguistic and semantic knowledge, has been observed. This limitation suggests that further exploration of different types of knowledge may be necessary to improve the performance of these models.

- Our systems still rely on large-scale annotated datasets for tasks like Sentiment Analysis and Multi-modal QA. This reliance poses challenges in terms of data collection, annotation, and scalability for real-world applications.

- The experiments presented in this thesis are primarily focused on English datasets while evaluation on non-English languages is missing. Moreover, the amount of datasets employed in some experiments is limited, such as the analysis of the bias of QA datasets, which is only conducted on two datasets. Therefore, experiments on more datasets including non-English data could be performed to further verify the effectiveness of our approaches.

- Although we have proposed techniques to incorporate multi-modal

information, there is still room for improvement in terms of effectively utilizing and integrating various types of multi-modal data, such as features of sensor data, to enhance the performance of Pre-trained Vision-Language Models.

Addressing these limitations will be crucial for the development of more robust and versatile language models capable of handling diverse real-world applications. We discuss some avenues for future work in the next section.

6.5 Future Work

6.5.1 Incorporating more diverse sources of external knowledge into PLMs

In this thesis, we proposed approaches in Chapter 3 for incorporating user/product preference into Sentiment Analysis and in Chapter 4 for injecting semantic and linguistic knowledge into Question Generation. However, there are more types of external knowledge needed to be integrated into PLMs. One of the limitations of PLMs is the deficiency of explicit world knowledge and common sense, resulting in suboptimal performance on certain tasks Bang et al. [2023], Lai et al. [2023]. Moreover, PLMs have to learn the knowledge of human preference in order to align with human preference on how tasks should be conducted [Ouyang et al., 2022] (user/product preference for Sentiment Analysis in Chapter 3 is a kind of human preference). By integrating this knowledge from various external sources (such as knowledge graphs as well as knowledge of human preference), PLMs could potentially demonstrate a more comprehensive, explainable worldview that is closer to humans.

Future work may involve exploration of more effective methods for diverse knowledge including world knowledge and common sense into PLMs such as ChatGPT. We could utilize ontologies or taxonomies to represent hierarchical relationships between concepts and employ semantic networks to indicate semantic relationships between concepts. One approach involves utilizing graph embeddings to represent

structured knowledge, where high-dimensional vectors are embedded to capture the semantic relationships between nodes in a knowledge graph. Moreover, for PLMs including ChatGPT and GPT-4, we might use approaches which inject knowledge into them without directly changing the architecture of such models [Min et al., 2022]. For example, we could softly inject such knowledge via carefully designing the prompts or the demonstration examples used in In-context Learning [Dong et al., 2022].

Another area of interest is investigating effective methods for aligning PLMs with human preferences [Ouyang et al., 2022]. While PLMs can leverage external knowledge sources to align with human preference (such as social bias, personalised preference, etc.), developing effective methods for this alignment remains a challenge. Future work may involve exploring different approaches beyond Reinforcement Learning from Human Feedback (RLHF) [Schulman et al., 2017, Stiennon et al., 2020] for training PLMs to better align with human preferences.

Furthermore, it is important to devise techniques to adapt external knowledge for domain-specific tasks, such as incorporating external knowledge like accounting for legal precedents or case law in the legal domain [Katz et al., 2023]. Moreover, PLMs should use external knowledge with additional information to adapt culture and behavior difference accordingly for certain tasks or questions when deploying PLMs for serving people in different countries [Shanahan, 2022, Dev et al., 2023, Santurkar et al., 2023].

6.5.2 Generating high-quality synthetic data for NLP tasks, especially in low-resource settings

In low-resource settings such as low-resource languages where annotated data is scarcely available, generating synthetic examples is a commonly used method for training a machine learning system [Nikolenko, 2021, He et al., 2022]. But creating high-quality synthetic data is still a challenging endeavor. Existing approaches include using generative models like Variational Autoencoders (VAEs) [Kingma

and Welling, 2013] or Generative Adversarial Networks (GANs) [Goodfellow et al., 2020] for synthetic data generation Abufadda and Mansour [2021]. Other methods can also be used in the creation of synthetic data. For example, in Chapter 4 we proposed an approach using heuristics with semantic and linguistic knowledge for generating synthetic data for Question Generation and Question Answering. Moreover, recent advancements in LLMs like ChatGPT and GPT-4 have made it possible to distill desirable data examples from such systems Ding et al. [2022], Wang et al. [2022], Wu et al. [2023b]. Nevertheless, generating high-quality synthetic data necessitates careful tuning of generative models, and comprehending the impact of various hyperparameters, architectures and even prompts used for querying LLMs Wu et al. [2023b] on the quality of generated data, is crucial for research.

Investigations could also delve into incorporating domain-specific knowledge into the data generation process. For instance, synthetic data in the medical domain could involve integrating medical terminology or knowledge from medical ontologies [Tsatsaronis et al., 2015, Naseem et al., 2022, Wu et al., 2022, Demner-Fushman et al., 2022], while the legal domain could involve integrating legal precedent or case law [Katz et al., 2023]. This incorporation of domain-specific knowledge could potentially enhance the synthetic data's quality and relevance to the target task or domain.

Moreover, the quality of synthetic data for different NLP tasks should be evaluated [Puri et al., 2020, Lyu et al., 2021]. As the usefulness of synthetic data can vary for different tasks or domains, so understanding its strengths and limitations could potentially foster the development of more effective NLP models.

6.5.3 Novel multi-modal fusion strategies for better integration of visual, acoustic, textual and other modality information in Video Question Answering and other multi-modal tasks

Video Question Answering (VQA) is a complex multi-modal task that necessitates the integration of visual, audio, and textual information to generate accurate

answers [Yang et al., 2003, Antol et al., 2015, Lei et al., 2018]. In Chapter 5, we proposed novel approaches incorporating semantic graph information and multi-grained alignment, resulting in improved performance for VideoQA systems. However, fusing multi-modal information can be challenging due to the unaligned nature of these modalities [Lee et al., 2022, Xiao et al., 2022b], and existing strategies may not suffice for achieving human-level performance. Moreover, there are other useful modalities that should be incorporated into VideoQA task as well as other multi-modal tasks. These include sensor data, depth maps, motion, and optical flow, which go beyond the commonly employed modalities of visual, acoustic, and textual information.

One potential area of future research is investigating various methods of fusing different modalities. Early fusion involves concatenating different modalities and passing them through a shared encoder, while late fusion processes modalities separately before combining them later. Hybrid approaches that integrate elements of early and late fusion might also prove effective. Additionally, developing techniques to handle missing or noisy modalities is essential [Lee et al., 2022], as real-world scenarios often present challenges when some modalities are missing or contain noise, making it difficult for VideoQA models to generate accurate answers. Attention mechanisms that weight the relevance of different modalities dynamically based on their relevance to the question can be employed, as well as techniques such as imputation or denoising to address missing or noisy modalities [Tran et al., 2017, Zhang et al., 2023].

Lastly, assessing the impact of different fusion strategies on both model performance and interpretability is crucial [Hu et al., 2021, Huang et al., 2022b]. Knowing the strengths and limitations of each strategy can inform the development of more powerful VideoQA models as well as other multi-modal systems.

Bibliography

Mohammad Abufadda and Khalid Mansour. A survey of synthetic data generation for machine learning. In *2021 22nd International Arab Conference on Information Technology (ACIT)*, pages 1–7. IEEE, 2021.

Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K. Marks, Chiori Hori, Peter Anderson, Stefan Lee, and Devi Parikh. Audio-visual scene-aware dialog. In *CVPR*, 2019.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1620. URL `https://www.aclweb.org/anthology/P19-1620`.

Reinald Kim Amplayo. Rethinking attribute representation and injection for sentiment classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5602–5613, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1562. URL `https://www.aclweb.org/anthology/D19-1562`.

Reinald Kim Amplayo, Jihyeok Kim, Sua Sung, and Seung-won Hwang. Cold-start aware user and product attention for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2535–2544, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1236. URL `https://www.aclweb.org/anthology/P18-1236`.

Reinald Kim Amplayo, Seonjae Lim, and Seung-won Hwang. Text length adaptation in sentiment classification. In *Asian Conference on Machine Learning*, pages 646–661. PMLR, 2019.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *ICCV*, 2015.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *International Conference on Learning Representations*, 2018.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. A clip-hitchhiker's guide to long video retrieval. *arXiv preprint arXiv:2205.08508*, 2022.

Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W05-0909`.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Emily M. Bender and Batya Friedman. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018. doi: 10.1162/tacl_a_00041. URL https://aclanthology.org/Q18-1041.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3: 1137–1155, 2003.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems*, 26, 2013.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E06-1032.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

Danqi Chen and Wen-tau Yih. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-tutorials.8. URL https://aclanthology.org/2020.acl-tutorials.8.

Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany, August 2016a. Association for Computational Linguistics. doi: 10.18653/v1/P16-1223. URL https://aclanthology.org/P16-1223.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL https://aclanthology.org/P17-1171.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 conference on Empirical Methods in Natural Language Processing*, pages 1650–1659,

Austin, Texas, November 2016b. Association for Computational Linguistics. doi: 10.18653/v1/D16-1171. URL https://www.aclweb.org/anthology/D16-1171.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7870–7881, 2020a.

Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *CVPR*, 2020b.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. Reinforcement learning based graph-to-sequence model for natural question generation. In *International Conference on Learning Representations*, 2019.

Cheng-Han Chiang and Hung-yi Lee. On the transferability of pre-trained language models: A study from artificial datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10518–10525, 2022.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the Sixth International Conference on Knowledge Capture*, pages 113–120, 2011.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://aclanthology.org/W19-4828.

Pedro Colon-Hernandez, Catherine Havasi, Jason Alonso, Matthew Huggins, and Cynthia Breazeal. Combining pre-trained language models and structured knowledge. *arXiv preprint arXiv:2101.12294*, 2021.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1269. URL `https://aclanthology.org/D18-1269`.

Jeff Da and Jungo Kasai. Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 1–12, 2019.

Long Hoang Dang, Thao Minh Le, Vuong Le, and Truyen Tran. Hierarchical object-oriented spatio-temporal reasoning for video question answering. *arXiv preprint arXiv:2106.13432*, 2021.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

Dina Demner-Fushman, Kevin Bretonnel Cohen, Sophia Ananiadou, and Junichi Tsujii, editors. *Proceedings of the 21st Workshop on Biomedical Language Processing*, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.bionlp-1.0`.

Sunipa Dev, Akshita Jha, Jaya Goyal, Dinesh Tewari, Shachi Dave, and Vinodkumar Prabhakaran. Building stereotype repositories with complementary approaches for scale and depth. In *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*, pages 84–90, Dubrovnik, Croatia, May 2023.

Association for Computational Linguistics. URL https://aclanthology.org/2023.c3nlp-1.9.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Kaustubh Dhole and Christopher D. Manning. Syn-QG: Syntactic and shallow semantic rules for question generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 752–765, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.69. URL https://www.aclweb.org/anthology/2020.acl-main.69.

Bosheng Ding, Chengwei Qin, Linlin Liu, Lidong Bing, Shafiq Joty, and Boyang Li. Is gpt-3 a good data annotator? *arXiv preprint arXiv:2212.10450*, 2022.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

Jianfeng Dong, Xirong Li, and Cees GM Snoek. Predicting visual features from text for image and video caption retrieval. *IEEE TMM*, 20(12):3377–3388, 2018.

Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. Dual encoding for video retrieval by text. *IEEE TPAMI*, 44(8):4065–4080, 2021.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32, 2019.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Zhendong Dong, Qiang Dong, and Changling Hao. HowNet and its computation of meaning. In *COLING 2010: Demonstrations*, pages 53–56, Beijing, China, August 2010. COLING 2010 Organizing Committee. URL `https://aclanthology.org/C10-3014`.

Cícero dos Santos and Maíra Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C14-1008`.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold,

Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Zi-Yi Dou. Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 521–526, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1054. URL https://www.aclweb.org/anthology/D17-1054.

Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1123. URL https://www.aclweb.org/anthology/P17-1123.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1090. URL https://www.aclweb.org/anthology/D17-1090.

Daria Dzendzik, Carl Vogel, and Jennifer Foster. English machine reading comprehension datasets: A survey. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2021.

Jeffrey L Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

Alexander Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. Template-based question generation from retrieved sentences for improved unsupervised question answering. In *Proceedings of the 58th Annual Meeting of*

the Association for Computational Linguistics, pages 4508–4513, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.413. URL https://www.aclweb.org/anthology/2020.acl-main.413.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.

Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *CVPR*, 2019.

Haytham M Fayek and Justin Johnson. Temporal reasoning via audio question answering. *IEEE TASLP*, 28:2283–2294, 2020.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*, 2019.

Danny Francis, Phuong Anh Nguyen, Benoit Huet, and Chong-Wah Ngo. Fusion of multimodal embeddings for ad-hoc video search. In *ICCVW*, 2019.

Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020.

Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, Jianfeng Gao, et al. Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, 14(3–4):163–352, 2022.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W18-2501`.

Matt Gardner, William Merrill, Jesse Dodge, Matthew E Peters, Alexis Ross, Sameer Singh, and Noah A Smith. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, 2021.

Mor Geva, Yoav Goldberg, and Jonathan Berant. Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1107. URL `https://aclanthology.org/D19-1107`.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney. Autotutor: an intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618, 2005. doi: 10.1109/TE.2005.856149.

Kyle Hamilton, Aparna Nayak, Bojan Božić, and Luca Longo. Is neuro-symbolic ai

meeting its promise in natural language processing? a structured review. *arXiv preprint arXiv:2202.12205*, 2022.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.207. URL https://aclanthology.org/2020.findings-emnlp.207.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1044. URL https://aclanthology.org/P17-1044.

Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. Generate, annotate, and learn: NLP with synthetic text. *Transactions of the Association for Computational Linguistics*, 10:826–842, 2022. doi: 10.1162/tacl_a_00492. URL https://aclanthology.org/2022.tacl-1.48.

Michael Heilman and Noah A Smith. Question generation via overgenerating transformations and ranking. Technical report, Carnegie-Mellon University, 2009.

Michael Heilman and Noah A. Smith. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, California, June 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N10-1086.

GE Hinton, JL McClelland, and DE Rumelhart. Distributed representations. In *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, pages 77–109. 1986.

Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in Neural Information Processing Systems*, 6, 1993.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, pages 44–51. Springer, 2011.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Giwon Hong, Junmo Kang, Doyeon Lim, and Sung-Hyon Myaeng. Handling anomalies of synthetic questions in unsupervised question answering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3441–3448, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.306. URL `https://aclanthology.org/2020.coling-main.306`.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL `https://doi.org/10.5281/zenodo.1212303`.

Wenxing Hu, Xianghe Meng, Yuntong Bai, Aiying Zhang, Gang Qu, Biao Cai, Gemeng Zhang, Tony W Wilson, Julia M Stephen, Vince D Calhoun, et al. Interpretable multimodal fusion networks reveal mechanisms of brain cognition. *IEEE Transactions on Medical Imaging*, 40(5):1474–1483, 2021.

Jingjia Huang, Yinan Li, Jiashi Feng, Xiaoshuai Sun, and Rongrong Ji. Clover:

Towards a unified video-language alignment and fusion model. *arXiv preprint arXiv:2207.07885*, 2022a.

Xiaoshui Huang, Wentao Qu, Yifan Zuo, Yuming Fang, and Xiaowei Zhao. Imfnet: Interpretable multimodal fusion for point cloud registration. *IEEE Robotics and Automation Letters*, 7(4):12323–12330, 2022b.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://aclanthology.org/P19-1356.

Tianbo Ji, Chenyang Lyu, Zhichao Cao, and Peng Cheng. Multi-hop question generation using hierarchical encoding-decoding and context switch mechanism. *Entropy*, 23(11):1449, 2021. doi: 10.3390/e23111449. URL https://doi.org/10.3390/e23111449.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://aclanthology.org/D17-1215.

Pin Jiang and Yahong Han. Reasoning with heterogeneous graph alignment for video question answering. In *AAAI*, 2020.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, December 2017. doi: 10.1162/tacl_a_00065. URL https://www.aclweb.org/anthology/Q17-1024.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL `https://www.aclweb.org/anthology/P17-1147`.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77, 2020.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*, 2020.

Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J Bommarito II. Natural language processing in the legal domain. *arXiv preprint arXiv:2302.12039*, 2023.

Junyeong Kim, Minuk Ma, Trung Pham, Kyungsu Kim, and Chang D Yoo. Modality shifting attention network for multi-modal video question answering. In *CVPR*, pages 10106–10115, 2020.

Nayoung Kim, Seong Jong Ha, and Je-Won Kang. Video question answering using language-guided deep compressed-domain video feature. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1708–1717, 2021.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*

(EMNLP), pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL https://www.aclweb.org/anthology/D14-1181.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Paul R Kingsbury and Martha Palmer. From treebank to propbank. In *LREC*, pages 1989–1993, 2002.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. Look at the first sentence: Position bias in question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1109–1121, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.84. URL https://aclanthology.org/2020.emnlp-main.84.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M.

Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, March 2019. doi: 10.1162/tacl_a_00276. URL `https://www.aclweb.org/anthology/Q19-1026`.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. *arXiv preprint arXiv:2304.05613*, 2023.

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *CVPR*, 2020.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Sangmin Lee, Sungjune Park, and Yong Man Ro. Audio-visual mismatch-aware video retrieval via association and adjustment. In *ECCV*, 2022.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. Tvqa: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1379, 2018.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021.

Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation,

and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1484. URL https://www.aclweb.org/anthology/P19-1484.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.86.

Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. Learning to answer questions in dynamic audio-visual scenarios. In *CVPR*, 2022a.

Guohao Li, Feng He, and Zhifan Feng. A CLIP-Enhanced method for video-language understanding. *arXiv:2110.07137*, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022b.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, August

2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353.

Xiangpeng Li, Lianli Gao, Xuanhan Wang, Wu Liu, Xing Xu, Heng Tao Shen, and Jingkuan Song. Learnable aggregating net with diversity learning for video question answering. In *ACMMM*, 2019a.

Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In *AAAI*, 2019b.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://aclanthology.org/C02-1150.

Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. W2vv++ fully deep learning for ad-hoc video search. In *ACMMM*, 2019c.

Yicong Li, Xiang Wang, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Invariant grounding for video question answering. In *CVPR*, 2022c.

Zhongli Li, Wenhui Wang, Li Dong, Furu Wei, and Ke Xu. Harvesting and refining question-answer pairs for unsupervised QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6719–6728, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.600. URL https://www.aclweb.org/anthology/2020.acl-main.600.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Ding Liu and Daniel Gildea. Semantic role features for machine translation. In

Proceedings of the 23rd International Conference on Computational Linguistics, pages 716–724, 2010.

Hongying Liu, Ruyi Luo, Fanhua Shang, Mantang Niu, and Yuanyuan Liu. Progressive semantic matching for video-text retrieval. In *ACMMM*, 2021a.

Linqing Liu, Patrick Lewis, Sebastian Riedel, and Pontus Stenetorp. Challenges in generalization in open domain question answering, 2021b.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of NAACL-HLT*, pages 1073–1094, 2019a.

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*, 2018.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908, 2020a.

Yang Liu, Guanbin Li, and Liang Lin. Cross-modal causal relational reasoning for event-level visual question answering. *arXiv preprint arXiv:2207.12647*, 2022.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020b. doi: 10.1162/tacl_a_00343. URL `https://aclanthology.org/2020.tacl-1.47`.

Yunfei Long, Mingyu Ma, Qin Lu, Rong Xiang, and Chu-Ren Huang. Dual memory network model for biased product review classification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 140–148, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6220. URL `https://www.aclweb.org/anthology/W18-6220`.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. *arXiv preprint arXiv:1606.00061*, 2016.

Yuanmao Luo, Ruomei Wang, Fuwei Zhang, Fan Zhou, and Shujin Lin. Temporal-aware mechanism with bidirectional complementarity for video q&a. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3273–3278. IEEE, 2022.

Chenyang Lyu, Jennifer Foster, and Yvette Graham. Improving document-level sentiment analysis with user and product context. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6724–6729, Barcelona, Spain (Online), December 2020a. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.590. URL `https://aclanthology.org/2020.coling-main.590`.

Chenyang Lyu, Tianbo Ji, and Yvette Graham. Incorporating context and knowledge for better sentiment analysis of narrative text. In Ricardo Campos, Alípio Mário Jorge, Adam Jatowt, and Sumit Bhatia, editors, *Proceedings of Text2Story - Third Workshop on Narrative Extraction From Texts co-located with 42nd European Conference on Information Retrieval, Text2Story@ECIR 2020, Lisbon, Portugal, April 14th, 2020 [online only]*, volume 2593 of *CEUR Workshop Proceedings*, pages

39–45. CEUR-WS.org, 2020b. URL http://ceur-ws.org/Vol-2593/paper5.pdf.

Chenyang Lyu, Lifeng Shang, Yvette Graham, Jennifer Foster, Xin Jiang, and Qun Liu. Improving unsupervised question answering via summarization-informed question generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4134–4148, 2021.

Chenyang Lyu, Jennifer Foster, and Yvette Graham. Extending the scope of out-of-domain: Examining QA models in multiple subdomains. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 24–37, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.insights-1.4. URL https://aclanthology.org/2022.insights-1.4.

Chenyang Lyu, Tianbo Ji, Yvette Graham, and Jennifer Foster. Is a video worth n n images? a highly efficient approach to transformer-based video question answering. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 183–189, Toronto, Canada (Hybrid), July 2023a. Association for Computational Linguistics. URL https://aclanthology.org/2023.sustainlp-1.12.

Chenyang Lyu, Tianbo Ji, Yvette Graham, and Jennifer Foster. Semantic-aware dynamic retrospective-prospective reasoning for event-level video question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 50–56, Toronto, Canada, July 2023b. Association for Computational Linguistics. URL https://aclanthology.org/2023.acl-srw.7.

Chenyang Lyu, Manh-Duy Nguyen, Van-Tu Ninh, Liting Zhou, Cathal Gurrin, and Jennifer Foster. Dialogue-to-video retrieval. In *ECIR*, 2023c.

Chenyang Lyu, Linyi Yang, Yue Zhang, Yvette Graham, and Jennifer Foster. Exploiting rich textual user-product context for improving personalized sentiment

analysis. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1419–1429, 2023d.

Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang, and Xu Sun. Cascading multiway attentions for document-level sentiment classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 634–643, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I17-1064.

Xiyao Ma, Qile Zhu, Yanlin Zhou, and Xiaolin Li. Improving question generation with sentence-level semantic matching and answer position inferring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8464–8471, 2020.

Avinash Madasu, Junier Oliva, and Gedas Bertasius. Learning to retrieve videos by asking questions. In *ACMMM*, 2022.

Sho Maeoki, Kohei Uehara, and Tatsuya Harada. Interactive video retrieval with dialog. In *CVPRW*, 2020.

Jianguo Mao, Wenbin Jiang, Xiangdong Wang, Zhifan Feng, Yajuan Lyu, Hong Liu, and Yong Zhu. Dynamic multistep reasoning based on video scene graph for video question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3894–3904, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.286. URL https://aclanthology.org/2022.naacl-main.286.

Lluís Màrquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. Special issue introduction: Semantic role labeling: An introduction to the special issue. *Computational Linguistics*, 34(2):145–159, 2008. doi: 10.1162/coli.2008.34.2.145. URL https://www.aclweb.org/anthology/J08-2001.

Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv:1804.02516*, 2018.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL `https://aclanthology.org/2022.emnlp-main.759`.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL `https://www.aclweb.org/anthology/D18-1206`.

Shashi Narayan, Gonçalo Simoes, Ji Ma, Hannah Craighead, and Ryan Mcdonald. Qurious: Question generation pretraining for text generation. *arXiv preprint arXiv:2004.11026*, 2020.

Usman Naseem, Adam G Dunn, Matloob Khushi, and Jinman Kim. Benchmarking for biomedical natural language processing tasks with a domain specific albert. *BMC bioinformatics*, 23(1):1–15, 2022.

Sergey I Nikolenko. *Synthetic data for deep learning*, volume 174. Springer, 2021.

OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2303.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

Martha Palmer, Daniel Gildea, and Nianwen Xue. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103, 2010.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://www.aclweb.org/anthology/P02-1040`.

Kyung-Mi Park, Young-Sook Hwang, and Hae Chang Rim. Two-phase semantic role labeling based on support vector machines. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 126–129, 2004.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL `https://www.aclweb.org/anthology/D14-1162`.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, 2019.

Gabriele Picco, Thanh Lam Hoang, Marco Luca Sbodio, and Vanessa Lopez. Neural unification for logic reasoning over natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3939–3950, 2021.

Barbara Plank. What to do about non-standard (or non-canonical) language in NLP. In Stefanie Dipper, Friedrich Neubarth, and Heike Zinsmeister, editors, *Proceedings of the 13th Conference on Natural Language Processing, KONVENS 2016, Bochum, Germany, September 19-21, 2016*, volume 16 of *Bochumer Linguistische Arbeitsberichte*, 2016. URL `https://www.linguistics.rub.de/konvens16/pub/2_konvensproc.pdf`.

Barbara Plank and Khalil Sima'an. Subdomain sensitive statistical parsing using raw corpora. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association, 2008. URL `http://www.lrec-conf.org/proceedings/lrec2008/summaries/120.html`.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August

2014. Association for Computational Linguistics. doi: 10.3115/v1/S14-2004. URL https://www.aclweb.org/anthology/S14-2004.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.468. URL https://www.aclweb.org/anthology/2020.emnlp-main.468.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26, 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *Technical Report*, 2019. URL https://insightcivic.s3.us-east-1.amazonaws.com/language-models.pdf.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://www.aclweb.org/anthology/D16-1264.

Ehud Reiter. A Structured Review of the Validity of BLEU. *Computational Linguistics*, 44(3):393–401, 09 2018. ISSN 0891-2017. doi: 10.1162/coli_a_00322. URL https://doi.org/10.1162/coli_a_00322.

Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. *Advances in Neural Information Processing Systems*, 28, 2015.

Anna Rogers. Changing the world by changing the data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2182–2194, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.170. URL https://aclanthology.org/2021.acl-long.170.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020.

Arpita Roy and Shimei Pan. Incorporating extra knowledge to enhance word embedding. In *IJCAI*, pages 4929–4935, 2020.

Ludan Ruan and Qin Jin. Survey: Transformer based video-language pre-training. *AI Open*, 3:1–13, 2022. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2022.01.001. URL https://www.sciencedirect.com/science/article/pii/S2666651022000018.

Sebastian Ruder and Avi Sil. Multi-domain multilingual question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language*

Processing: Tutorial Abstracts, pages 17–21, Punta Cana, Dominican Republic & Online, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-tutorials.4.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Arka Sadhu, Kan Chen, and Ram Nevatia. Video question answering with phrases via semantic roles. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2460–2478, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.196. URL https://aclanthology.org/2021.naacl-main.196.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1156. URL https://www.aclweb.org/anthology/P18-1156.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? *arXiv preprint arXiv:2303.17548*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sebastian Schuster, Ranjay Krishna, Angel Chang, Li Fei-Fei, and Christopher D. Manning. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *EMNLPW*, 2015.

Priyanka Sen and Amir Saffari. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing (EMNLP), pages 2429–2438, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.190. URL https://aclanthology.org/2020.emnlp-main.190.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 1715. Association for Computational Linguistics, 2016.

Ahjeong Seo, Gi-Cheon Kang, Joonhan Park, and Byoung-Tak Zhang. Attend what you need: Motion-appearance synergistic networks for video question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6167–6177, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.481. URL https://aclanthology.org/2021.acl-long.481.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.439. URL https://www.aclweb.org/anthology/2020.emnlp-main.439.

Murray Shanahan. Talking about large language models. *arXiv preprint arXiv:2212.03551*, 2022.

Peng Shi and Jimmy J. Lin. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255, 2019.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D13-1170`.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832, Online, April 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.eacl-main.156`.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936. PMLR, 2019.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019a.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939, Brussels, Belgium, October-November 2018. Association for Computational

Linguistics. doi: 10.18653/v1/D18-1427. URL https://www.aclweb.org/anthology/D18-1427.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019b.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Alon Talmor and Jonathan Berant. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1485. URL https://aclanthology.org/P19-1485.

Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, Lisbon, Portugal, September 2015a. Association for Computational Linguistics. doi: 10.18653/v1/D15-1167. URL https://www.aclweb.org/anthology/D15-1167.

Duyu Tang, Bing Qin, and Ting Liu. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1014–1023, Beijing, China, July 2015b. Association for Computational Linguistics. doi: 10.3115/v1/P15-1098. URL https://www.aclweb.org/anthology/P15-1098.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, pages 4631–4640, 2016.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL `https://aclanthology.org/P19-1452`.

Luan Tran, Xiaoming Liu, Jiayu Zhou, and Rong Jin. Missing modalities imputation via cascaded residual autoencoder. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1405–1414, 2017.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2623. URL `https://www.aclweb.org/anthology/W17-2623`.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16(1):1–28, 2015.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1008. URL `https://aclanthology.org/P16-1008`.

Dusan Varis and Ondřej Bojar. Sequence length is a domain: Length-based overfitting in transformer models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8246–8257, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.650. URL https://aclanthology.org/2021.emnlp-main.650.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

G Vinodhini and RM Chandrasekaran. Sentiment analysis and opinion mining: a survey. *International Journal*, 2(6):282–292, 2012.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Ge Yuying, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. In *CVPR*, 2023.

Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. On position embeddings in bert. In *International Conference on Learning Representations*, 2020.

Jianyu Wang, Bing-Kun Bao, and Changsheng Xu. Dualvgr: A dual-visual graph reasoning unit for video question answering. *IEEE Transactions on Multimedia*, 24:3369–3380, 2021a.

Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer.(2017). In *ICLR 2017: International Conference on Learning Representations, Toulon, France, April 24-26: Proceedings*, pages 1–15.

Xiaohan Wang, Linchao Zhu, and Yi Yang. T2VLAD: global-local sequence alignment for text-video retrieval. In *CVPR*, 2021b.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021c.

Yequan Wang, Minlie Huang, Li Zhao, et al. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on Empirical Methods in Natural Language Processing*, pages 606–615, 2016.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

Jiwei Wei, Yang Yang, Xing Xu, Xiaofeng Zhu, and Heng Tao Shen. Universal weighting metric learning for cross-modal retrieval. *IEEE TPAMI*, 44(10):6534–6545, 2021a.

Xiaokai Wei, Shen Wang, Dejiao Zhang, Parminder Bhatia, and Andrew Arnold. Knowledge enhanced pretrained language models: A compreshensive survey. *arXiv preprint arXiv:2110.08455*, 2021b.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie

Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *CVPR*, 2019.

Honghan Wu, Minhong Wang, Jinge Wu, Farah Francis, Yun-Hsuan Chang, Alex Shavick, Hang Dong, Michael TC Poon, Natalie Fitzpatrick, Adam P Levine, et al. A survey on clinical natural language processing in the united kingdom from 2007 to 2022. *NPJ digital medicine*, 5(1):186, 2022.

Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long, et al. Graph neural networks for natural language processing: A survey. *FTML*, 16(2):119–328, 2023a.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *CoRR*, abs/2304.14402, 2023b. URL `https://arxiv.org/abs/2304.14402`.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Zhen Wu, Xin-Yu Dai, Cunyan Yin, Shujian Huang, and Jiajun Chen. Improving review representations with user attention and product attention for sentiment classification. *CoRR*, abs/1801.07861, 2018. URL `http://arxiv.org/abs/1801.07861`.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *TNNLS*, 32(1):4–24, 2020.

Junbin Xiao, Xindi Shang, Xun Yang, Sheng Tang, and Tat-Seng Chua. Visual relation grounding in videos. In *ECCV*, 2020.

Junbin Xiao, Angela Yao, Zhiyuan Liu, Yicong Li, Wei Ji, and Tat-Seng Chua. Video as conditional graph hierarchy for multi-granular question answering. In *AAAI*, volume 36, pages 2804–2812, 2022a.

Shaoning Xiao, Long Chen, Kaifeng Gao, Zhao Wang, Yi Yang, Zhimeng Zhang, and Jun Xiao. Rethinking multi-modal alignment in multi-choice VideoQA from feature and sample perspectives. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8188–8198, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. URL `https://aclanthology.org/2022.emnlp-main.561`.

Yuxi Xie, Liangming Pan, Dongzhe Wang, Min-Yen Kan, and Yansong Feng. Exploring question-specific rewards for generating deep questions. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2534–2546, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.228. URL `https://www.aclweb.org/anthology/2020.coling-main.228`.

Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Yanghua Xiao. Cn-dbpedia: A never-ending chinese knowledge extraction system. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 428–438. Springer, 2017.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *EMNLP*, 2021a.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5288–5296, 2016.

Li Xu, He Huang, and Jun Liu. Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9878–9888, 2021b.

Chenghao Yang and Xuezhe Ma. Improving stability of fine-tuning pretrained language models via component-wise gradient norm clipping. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4854–4859, 2022.

Hui Yang, Lekha Chaisorn, Yunlong Zhao, Shi-Yong Neo, and Tat-Seng Chua. Videoqa: question answering on news video. In *ACMMM*, 2003.

Pinci Yang, Xin Wang, Xuguang Duan, Hong Chen, Runze Hou, Cong Jin, and Wenwu Zhu. Avqa: A dataset for audio-visual question answering on videos. In *ACMMM*, 2022.

Xun Yang, Jianfeng Dong, Yixin Cao, Xun Wang, Meng Wang, and Tat-Seng Chua. Tree-augmented cross-modal encoding for complex-query video retrieval. In *SIGIR*, 2020a.

Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. Bert representations for video question answering. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1556–1565, 2020b.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32, 2019.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

Qinghao Ye, Guohai Xu, Ming Yan, Haiyang Xu, Qi Qian, Ji Zhang, and Fei Huang. Hitea: Hierarchical temporal-aware video-language pre-training. *arXiv preprint arXiv:2212.14546*, 2022.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in Neural Information Processing Systems*, 31, 2018.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv preprint arXiv:2010.00796*, 2020.

Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 471–487, 2018.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *CVPR*, 2019.

Zhigang Yuan, Fangzhao Wu, Junxin Liu, Chuhan Wu, Yongfeng Huang, and Xing Xie. Neural review rating prediction with user and product memory. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7,2019*, pages 2341–2344, 2019.

Heeseung Yun, Youngjae Yu, Wonsuk Yang, Kangil Lee, and Gunhee Kim. Pano-avqa: Grounded audio-visual question answering on 360deg videos. In *ICCV*, 2021.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. MERLOT: Multimodal neural script knowledge models. In *NeurIPS*, 2021.

Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In *SIGIR*, pages 26–32, 2003.

Fuwei Zhang, Ruomei Wang, Fan Zhou, and Yuanmao Luo. Erm: Energy-based refined-attention mechanism for video question answering. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

Jipeng Zhang, Jie Shao, Rui Cao, Lianli Gao, Xing Xu, and Heng Tao Shen. Action-centric relation transformer network for video question answering. *IEEE TCSVT*, 32(1):63–74, 2020a.

Shiyue Zhang and Mohit Bansal. Addressing semantic drift in question generation for semi-supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2495–2509, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1253. URL `https://www.aclweb.org/anthology/D19-1253`.

Xi Zhang, Feifei Zhang, and Changsheng Xu. Explicit cross-modal representation learning for visual commonsense reasoning. *IEEE Transactions on Multimedia*, 24: 2986–2997, 2021a.

You Zhang, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. MA-BERT: Learning representation by incorporating multi-attribute knowledge in transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2338–2343, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.206. URL `https://aclanthology.org/2021.findings-acl.206`.

Yue Zhang, Chengtao Peng, Qiuli Wang, Dan Song, Kaiyan Li, and S Kevin Zhou. Unified multi-modal image synthesis for missing modality imputation. *arXiv preprint arXiv:2304.05340*, 2023.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL `https://aclanthology.org/P19-1139`.

Zhuosheng Zhang, Hai Zhao, and Rui Wang. Machine reading comprehension: The role of contextualized language models and beyond. *arXiv preprint arXiv:2005.06249*, 2020b.

Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, and Yueting Zhuang. Video question answering via hierarchical spatio-temporal attention networks. In *IJCAI*, 2017.

Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: datasets, algorithms and challenges. *arXiv preprint arXiv:2203.01225*, 2022a.

Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. *arXiv preprint arXiv:2203.01225*, 2022b.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of the 2016 conference on Empirical Methods in Natural Language Processing*, pages 247–256, 2016.

Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27, 2015.

Zihao Zhu. From shallow to deep: Compositional reasoning over graphs for visual question answering. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8217–8221. IEEE, 2022.

Appendix A

Question Answering

A.1 A Novel Approach to Question Generation

A.1.1 Generated QA Examples

More Wikipedia-based <*passage, answer, question*> examples generated by our BART-QG model are shown in Table A.1, Table A.2 and Table A.3.

A.2 Extending the Scope of Out-of-Domain: Examining QA models in multiple subdomains

A.2.1 Average Text Length and Answer Position for All Question Types

We show the average text length of *context, question and answer* as well as the average answer position on character-level, word-level and sentence-level for QA examples in all question types in SQuAD1.1 and NewsQA in Table A.4 and Table A.5.

A.2.2 Question Type Proportions, Average Text Length and Average Answer Position for *Long* and *Short* Text Length

The median of the *context, question, answer* is shown in Table A.6. We show the question type proportion, average text length for *context, question and answer* as well as the average answer position on character-level, word-level and sentence-level for QA examples in *long* and *short* groups of *context, question, answer* in SQuAD1.1

Passage	Answer	Question
At a professional level, most matches produce only a few goals. For example, the 2005–06 season of the English Premier League produced an average of 2.48 goals per match. The Laws of the Game do not specify any player positions other than goalkeeper, but a number of specialised roles have evolved.	the 2005–06 season	when did the english football team produce an average of 2.49 goals per match , according to the laws of the game ?
The Hebrew Book Week is held each June and features book fairs, public readings, and appearances by Israeli authors around the country. During the week, Israel's top literary award, the Sapir Prize, is presented.	The Hebrew Book Week	what is held every june to celebrate the publication of books in hebrew ?
On December 12, 2016, Senate Majority Leader Republican Mitch McConnell expressed confidence in U.S. intelligence. McConnell added that investigation of Russia's actions should be bipartisan and held by the Senate Intelligence Committee. The next day, Senate Intelligence Committee Chairman Richard Burr (R-NC) and Vice Chairman Mark Warner (D-VA) announced the scope of the committee's .	Republican Mitch McConnell	which republican has called for a special committee to investigate russia 's alleged meddling in the 2016 presidential election ?
Meanwhile, the Soho Mint struck coins for the East India Company, Sierra Leone and Russia, while producing high-quality planchets, or blank coins, to be struck by national mints elsewhere. The firm sent over 20 million blanks to Philadelphia, to be struck into cents and half-cents by the United States Mint —Mint Director Elias Boudinot found them to be "perfect and beautifully polished".	Elias Boudinot	who has been working for a company that made coins for the us mint ?

Table A.1: Some generated QA examples.

Passage	Answer	Question
In March 2008 as part of the annual budget, the government introduced several laws to amend the Immigration and Refugee Protection Act. The changes would have helped to streamline immigrant application back-up, to speed up application for skilled workers and to rapidly reject other ones that are judged not admissible by immigration officers. Immigrant applications had risen to a high of 500,000, creating a delay of up to six months for an application to be processed.	March 2008	when did the uk introduce new immigration laws ?
The other group members as far back as 1996 had noticed Paddy Clancy's unusual mood swings. In the spring of 1998 the cause was finally detected; Paddy had a brain tumor as well as lung cancer. His wife waited to tell him about the lung cancer, so as not to discourage him when he had a brain operation.	the spring of 1998	in what time was paddy diagnosed with lung cancer ?
In 1365 officials were created to supervise the fish market in the town, whilst illegal fishing and oyster cultivation was targeted by the bailiffs in an edict from 1382, which prohibited the forestalling of fish by blocking the river, the dredging of oysters out of season and the obstructing of the river. Colchester artisans included clockmakers, who maintained clocks in church towers across north Essex and Suffolk.	north Essex	where were hundreds of clocks made by local artisans ?
Badge numbers for Sheriffs and Deputies consist of a prefix number, which represents the county number, followed by a one to three digit number, which represents the Sheriff's or Deputy's number within that specific office. The Sheriff's badge number in each county is always #1. So the Sheriff from Bremer County would have an ID number of 9-1 (9 is the county number for Bremer County and 1 is the number for the Sheriff).	The Sheriff's badge number	what is the number used to identify the sheriff in each county ?

Table A.2: Some generated QA examples.

Passage	Answer	Question
Appian wrote that Calpurnius Piso was sent as a commander to Hispania because there were revolts. The following year Servius Galba was sent without soldiers because the Romans were busy with Cimbrian War and a slave rebellion in Sicily (the [Third Servile War], 104-100 BC). In the former war the Germanic tribes of the Cimbri and the Teutones migrated around Europe and invaded territories of allies of Rome, particularly in southern France, and routed the Romans in several battles until their final defeat.	Calpurnius Piso	who was sent to the south of italy to fight for the roman empire ?
The parish churches of Sempringham, Birthorpe, Billingborough, and Kirkby were already appropriated. Yet in 1247, Pope Innocent IV granted to the master the right to appropriate the church of Horbling, because there were 200 women in the priory who often lacked the necessaries of life. The legal expenses of the order at the papal curia perhaps accounted for their poverty.	200	there were how many women in the priory of horbling in the 12th century ?
"Jerry West is the reason I came to the Lakers", O'Neal later said. They used their 24th pick in the draft to select Derek Fisher. During the 1996–97 season, the team traded Cedric Ceballos to Phoenix for Robert Horry. O'Neal led the team to a 56–26 record, their best effort since 1990–91, despite missing 31 games due to a knee injury. O'Neal averaged 26.2 ppg and 12.5 rpg and finished third in the league in blocked shots (2.88 bpg) in 51 games.	the 1996–97 season	when do the phoenix suns begin with a trade to the los angeles clippers ?
Finnish popular music also includes various kinds of dance music; tango, a style of Argentine music, is also popular. One of the most productive composers of popular music was Toivo Kärki, and the most famous singer Olavi Virta (1915–1972). Among the lyricists, Sauvo Puhtila (1928–2014), Reino Helismaa (died 1965) and Veikko "Vexi" Salmi are a few of the most notable writers. The composer and bandleader Jimi Tenor is well known for his brand of retro-funk music.	Reino Helismaa	who has been hailed as one of finland 's most important writers ?

Table A.3: Some generated QA examples.

		Context	Question	Answer
SQuAD1.1	HUM	123.20	9.79	2.82
	LOC	117.18	9.62	2.78
	DESC	119.32	9.91	5.82
	ENTY	117.43	10.54	3.04
	NUM	121.09	10.11	2.08
NewsQA	HUM	495.79	6.55	2.82
	LOC	478.84	6.34	2.87
	DESC	513.00	6.25	7.62
	ENTY	505.94	6.76	4.27
	NUM	476.23	7.20	2.07

Table A.4: The average text length of context, question and answer in QA examples of each question type in the SQuAD1.1 and NewsQA training data.

		Char-Level	Word-Level	Sent-Level
SQuAD1.1	HUM	317.85	54.90	1.61
	LOC	308.81	53.71	1.53
	DESC	342.97	60.00	1.79
	ENTY	317.75	55.16	1.63
	NUM	315.78	56.19	1.67
NewsQA	HUM	532.11	101.02	3.71
	LOC	566.02	107.99	3.95
	DESC	844.13	160.05	5.98
	ENTY	751.48	143.90	5.49
	NUM	763.73	145.26	5.47

Table A.5: The average answer position of character-level, word-level and sentence-level in QA examples of each question type in the SQuAD1.1 and NewsQA training data.

	Context	Question	Answer
SQuAD1.1	110	10	2
NewsQA	534	6	2

Table A.6: The median of the *context, question, answer* length used to partition *long* and *short* subdomains.

		LOC	ENTY	HUM	NUM	DESC
SQuAD1.1	Long	11.11	26.68	21.65	24.8	15.43
	Short	11.73	28.42	19.68	24.2	15.52
NewsQA	Long	10.4	18.08	29.94	16.81	24.71
	Short	12.38	15.86	30.24	20.9	20.55

Table A.7: The percentage of each question type in *long context* and *short context* groups.

		LOC	ENTY	HUM	NUM	DESC
SQuAD1.1	Long	10.36	28.59	20.37	24.73	15.63
	Short	12.11	26.90	20.84	24.35	15.37
NewsQA	Long	9.45	18.29	29.70	23.66	18.90
	Short	12.96	15.91	30.40	14.98	25.63

Table A.8: The percentage of each question type in *long question* and *short question* groups.

		LOC	ENTY	HUM	NUM	DESC
SQuAD1.1	Long	10.87	27.32	19.69	21.8	19.86
	Short	11.79	27.72	21.29	26.29	12.55
NewsQA	Long	9.37	19.87	23.16	9.31	38.17
	Short	13.13	14.48	36.03	27.05	9.29

Table A.9: The percentage of each question type in *long answer* and *short answer* groups.

		Context	Question	Answer
SQuAD1.1	Long	84.53	9.99	3.09
	Short	155.88	10.14	3.23
NewsQA	Long	350.44	6.54	3.79
	Short	641.35	6.69	4.25

Table A.10: The average answer position on character-level, word-level and sentence-level in QA examples of *long context* and *short context* groups.

		Context	Question	Answer
SQuAD1.1	Long	119.12	7.8	3.25
	Short	120.76	13.57	3.03
NewsQA	Long	491.15	4.96	4.45
	Short	501.55	8.66	3.49

Table A.11: The average answer position on character-level, word-level and sentence-level in QA examples of *long question* and *short question* groups.

		Context	Question	Answer
SQuAD1.1	Long	119.08	10.18	1.42
	Short	120.79	9.88	5.77
NewsQA	Long	489.32	6.82	1.5
	Short	503.34	6.37	6.95

Table A.12: The average answer position on character-level, word-level and sentence-level in QA examples of *long answer* and *short answer* groups.

		Char	Word	Sent
SQuAD1.1	Long	402.02	70.36	2.14
	Short	239.75	41.78	1.17
NewsQA	Long	864.85	165.73	6.40
	Short	510.58	95.94	3.37

Table A.13: The average answer position on character-level, word-level and sentence-level in QA examples of *long context* and *short context* groups.

		Char	Word	Sent
SQuAD1.1	Long	342.02	59.70	1.74
	Short	305.65	53.45	1.58
NewsQA	Long	726.78	138.64	5.22
	Short	655.98	124.50	4.61

Table A.14: The average answer position on character-level, word-level and sentence-level in QA examples of *long question* and *short question* groups.

		Char	Word	Sent
SQuAD1.1	Long	324.65	57.77	1.71
	Short	316.70	54.65	1.60
NewsQA	Long	795.46	150.20	5.61
	Short	595.00	114.17	4.26

Table A.15: The average answer position on character-level, word-level and sentence-level in QA examples of *long answer* and *short answer* groups.

	Char	Word	Sent
SQuAD1.1	262	46	1
NewsQA	358	67	2

Table A.16: The median of the answer position on character-level, word-level and sentence-level used to partition *front* and *back* subdomains.

		LOC	ENTY	HUM	NUM	DESC
SQuAD1.1	Front	11.74	27.8	20.25	24.97	14.81
	Back	11.11	27.32	21.06	24.02	16.14
NewsQA	Front	13.07	15.59	37.2	15.61	18.46
	Back	9.71	18.36	22.97	22.1	26.8

Table A.17: The percentage of each question type in *front* and *back* groups on character-level answer position

and NewsQA in Table A.7, Table A.8, Table A.9, Table A.10 Table A.11, Table A.12, Table A.13, Table A.14, Table A.15.

A.2.3 Question Type Proportions, Average Text Length and Average Answer Position for QA examples with *Front* and *Back* Answer Positions

The median of the answer position on character-level, word-level and sentence-level is shown in Table A.16. We show the question type proportion, average text length for *context, question and answer* as well as the average answer position on character-level, word-level and sentence-level for QA examples in *front* and *back* groups of answer positions in character-level, word-level and sentence-level in SQuAD1.1 and NewsQA in Table A.17, Table A.18, Table A.19, Table A.20, Table A.21, Table A.22, Table A.23, Table A.24, Table A.25.

		LOC	ENTY	HUM	NUM	DESC
SQuAD1.1	Front	11.76	28.05	20.28	24.49	14.99
	Back	11.16	27.08	21.00	24.45	15.94
NewsQA	Front	13.02	15.59	37.20	15.64	18.48
	Back	9.74	18.43	22.85	22.11	26.81

Table A.18: The percentage of each question type in *front* and *back* groups on word-level answer position

		LOC	ENTY	HUM	NUM	DESC
SQuAD1.1	Front	11.72	27.83	20.60	24.48	14.95
	Back	11.04	27.18	20.71	24.56	16.15
NewsQA	Front	13.19	15.76	36.08	16.36	18.54
	Back	9.56	18.54	23.11	22.06	26.67

Table A.19: The percentage of each question type in *front* and *back* groups on sentence-level answer position

		Char	Word	Sent
SQuAD1.1	Front	116.25	20.6	0.44
	Back	524.15	91.3	2.85
NewsQA	Front	145.24	28.72	0.61
	Back	1230.24	232.96	9.15

Table A.20: The average answer position on character-level, word-level and sentence-level in QA examples of *front* and *back* groups of character-level answer position.

		Char	Word	Sent
SQuAD1.1	Front	127.4	19.34	0.44
	Back	515.71	93.09	2.88
NewsQA	Front	151.46	28.04	0.65
	Back	1229.77	234.74	9.17

Table A.21: The average answer position on character-level, word-level and sentence-level in QA examples of *front* and *back* groups of word-level answer position.

		Char	Word	Sent
SQuAD1.1	Front	158.46	26.12	0.4
	Back	532.52	95.11	3.28
NewsQA	Front	183.56	35.56	0.63
	Back	1280.56	242.86	9.89

Table A.22: The average answer position on character-level, word-level and sentence-level in QA examples of *front* and *back* groups of sentence-level answer position.

		Context	Question	Answer
SQuAD1.1	Front	108.80	9.83	3.06
	Back	130.77	10.30	3.26
NewsQA	Front	473.52	6.50	3.28
	Back	518.08	6.72	4.75

Table A.23: The average text length of context, question and answer in QA examples of *front* and *back* groups of character-level answer position

		Context	Question	Answer
SQuAD1.1	Front	109.21	9.84	3.03
	Back	130.50	10.28	3.30
NewsQA	Front	473.13	6.50	3.32
	Back	518.72	6.72	4.72

Table A.24: The average text length of context, question and answer in QA examples of *front* and *back* groups of word-level answer position

		Context	Question	Answer
SQuAD1.1	Front	110.14	9.93	3.04
	Back	132.44	10.23	3.33
NewsQA	Front	474.28	6.52	3.58
	Back	521.11	6.73	4.54

Table A.25: The average text length of context, question and answer in QA examples of *front* and *back* groups of sentence-level answer position

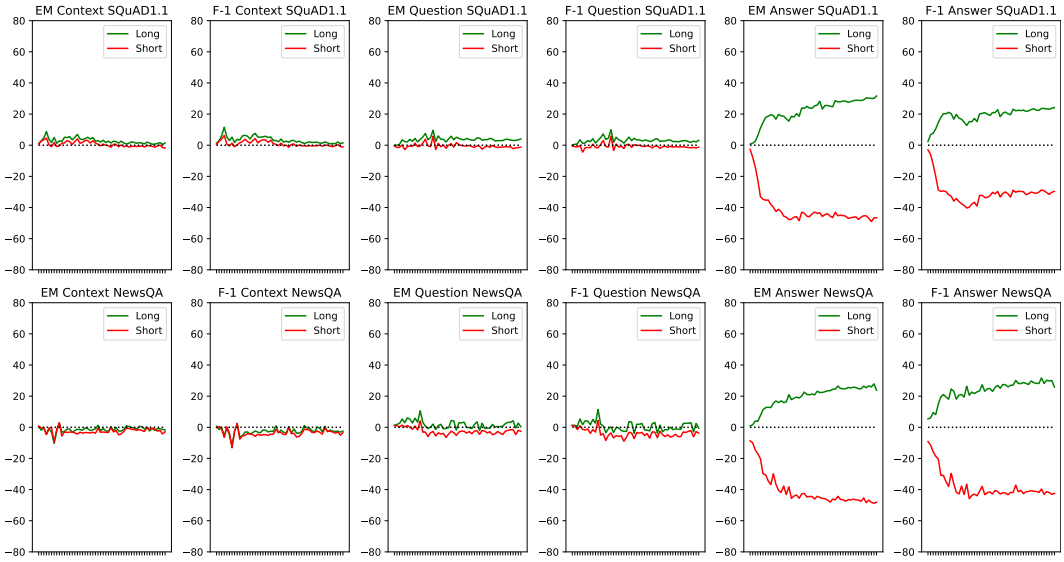

Figure A.1: Visualization of performance (EM and F-1 score) difference curves over *short* and *long* context, question and answer (from left to right) on SQuAD1.1 (top) and NewsQA (bottom). The *green, red* lines represent the difference of the performance on *long group* and *short group*. The dashed line is 0, indicating that two QA systems have the same performance. When the sample size increases, curves in *context* and *question* length converge to the dashed line, whereas there are substantial differences in the performance of QA_L and QA_S in *answer length* subdomain.

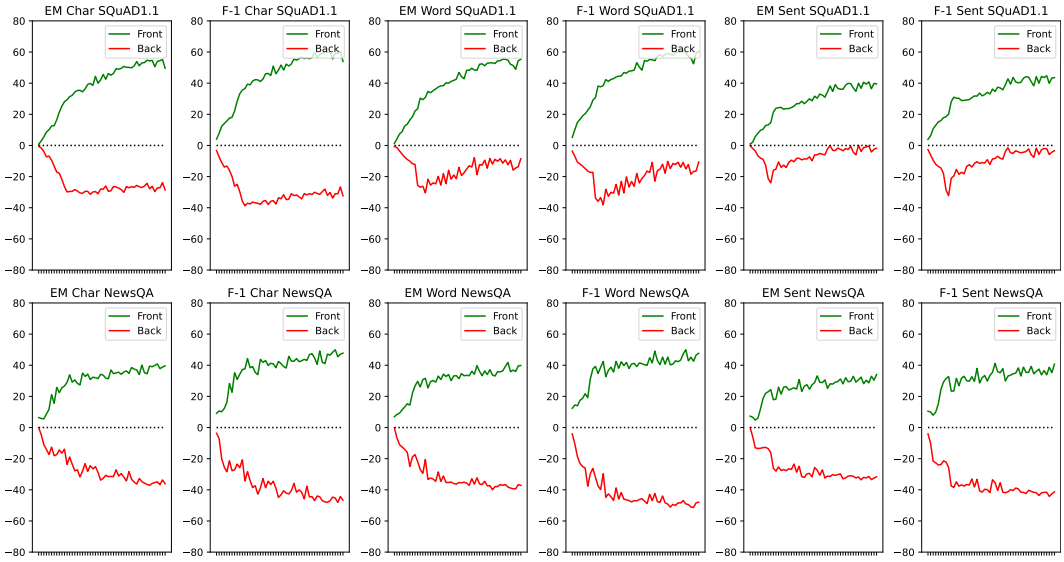

Figure A.2: Visualization of performance (EM and F-1 score) difference curves over *front* and *back* answer positions (char-level, word-level and sentence-level from left to right) on SQuAD1.1 (top) and NewsQA (bottom). The *green, red* lines represent the difference of the performance on *front group* and *back group*. The dashed line is 0, indicating that two QA systems have the same performance. The curves show that there are substantially difference in the performance of QA_F and QA_B in *answer position* subdomains especially for character-level and word-level answer positions.

A.2.4 QA examples with *long* answers and *short* answers

We give some QA examples with *long* answers and *short* answers in Table A.26 and Table A.27.

A.2.5 QA examples with *front* answers and *back* answers

We give some QA examples with character-level answer positions in *front* group and *back* group in Table A.28 and Table A.29.

A.2.6 Performance Difference for Text Length and Answer Position Experiments

We also show the difference of the performance (EM and F-1 score) between QA systems ($QA_L - QA_S$ and $QA_F - QA_B$) on subdomains of *text length* and *answer position* in Figure A.1 and Figure A.2.

Answer Length	Question	Context
Long	Where was the main focus of Pan-Slavism?	Pan-Slavism, a movement which came into prominence in the mid-19th century, emphasized the common heritage and unity of all the Slavic peoples. The main focus was in the Balkans where the South Slavs had been ruled for centuries by other empires: ***the Byzantine Empire, Austria-Hungary, the Ottoman Empire, and Venice***. The Russian Empire used Pan-Slavism as a political tool; as did the Soviet Union, which gained political-military influence and control over most Slavic-majority nations between 1945 and 1948 and retained a hegemonic role until the period 1989–1991.
Long	What is one reason for homologs to appear?	Genes with a most recent common ancestor, and thus a shared evolutionary ancestry, are known as homologs. These genes appear either from ***gene duplication within an organism's genome***, where they are known as paralogous genes, or are the result of divergence of the genes after a speciation event, where they are known as orthologous genes,:7.6 and often perform the same or similar functions in related organisms. It is often assumed that the functions of orthologous genes are more similar than those of paralogous genes, although the difference is minimal.
Long	How does the water vapor that rises in warm air turn into clouds?	Solar radiation is absorbed by the Earth's land surface, oceans – which cover about 71% of the globe – and atmosphere. Warm air containing evaporated water from the oceans rises, causing atmospheric circulation or convection. ***When the air reaches a high altitude, where the temperature is low, water vapor condenses into clouds***, which rain onto the Earth's surface, completing the water cycle. The latent heat of water condensation amplifies convection, producing atmospheric phenomena such as wind, cyclones and anti-cyclones. Sunlight absorbed by the oceans and land masses keeps the surface at an average temperature of 14 °C. By photosynthesis green plants convert solar energy into chemically stored energy, which produces food, wood and the biomass from which fossil fuels are derived.

Table A.26: Examples of QA examples with *long* answers where answers are highlighted.

Answer Length	Question	Context
Short	Who led the Exodus?	According to the Hebrew Bible narrative, Jewish ancestry is traced back to the Biblical patriarchs such as Abraham, Isaac and Jacob, and the Biblical matriarchs Sarah, Rebecca, Leah, and Rachel, who lived in Canaan around the 18th century BCE. Jacob and his family migrated to Ancient Egypt after being invited to live with Jacob's son Joseph by the Pharaoh himself. The patriarchs' descendants were later enslaved until the Exodus led by ***Moses***, traditionally dated to the 13th century BCE, after which the Israelites conquered Canaan.
Short	When did the Duke of Kent die?	Victoria was the daughter of Prince Edward, Duke of Kent and Strathearn, the fourth son of King George III. Both the Duke of Kent and King George III died in ***1820***, and Victoria was raised under close supervision by her German-born mother Princess Victoria of Saxe-Coburg-Saalfeld. She inherited the throne aged 18, after her father's three elder brothers had all died, leaving no surviving legitimate children. The United Kingdom was already an established constitutional monarchy, in which the sovereign held relatively little direct political power. Privately, Victoria attempted to influence government policy and ministerial appointments; publicly, she became a national icon who was identified with strict standards of personal morality.
Short	What is the evaluator called in a breed show?	In conformation shows, also referred to as breed shows, ***a judge*** familiar with the specific dog breed evaluates individual purebred dogs for conformity with their established breed type as described in the breed standard. As the breed standard only deals with the externally observable qualities of the dog (such as appearance, movement, and temperament), separately tested qualities (such as ability or health) are not part of the judging in conformation shows.

Table A.27: Examples of QA examples with *short* answers where answers are highlighted.

Answer Position	Question	Context
Front	What are the first names of the men that invented youtube?	According to a story that has often been repeated in the media, ***Hurley and Chen*** developed the idea for YouTube during the early months of 2005, after they had experienced difficulty sharing videos that had been shot at a dinner party at Chen's apartment in San Francisco. Karim did not attend the party and denied that it had occurred, but Chen commented that the idea that YouTube was founded after a dinner party ẅas probably very strengthened by marketing ideas around creating a story that was very digestible.
Front	Who became Chairman of the Council of Ministers in 1985?	In the fall of 1985, Gorbachev continued to bring younger and more energetic men into government. On September 27, ***Nikolai Ryzhkov*** replaced 79-year-old Nikolai Tikhonov as Chairman of the Council of Ministers, effectively the Soviet prime minister, and on October 14, Nikolai Talyzin replaced Nikolai Baibakov as chairman of the State Planning Committee (GOSPLAN). At the next Central Committee meeting on October 15, Tikhonov retired from the Politburo and Talyzin became a candidate. Finally, on December 23, 1985, Gorbachev appointed Yeltsin First Secretary of the Moscow Communist Party replacing Viktor Grishin.
Front	During what seasons is fog common in Boston?	Fog is fairly common, particularly in ***spring and early summer***, and the occasional tropical storm or hurricane can threaten the region, especially in late summer and early autumn. Due to its situation along the North Atlantic, the city often receives sea breezes, especially in the late spring, when water temperatures are still quite cold and temperatures at the coast can be more than 20 °F (11 °C) colder than a few miles inland, sometimes dropping by that amount near midday. Thunderstorms occur from May to September, that are occasionally severe with large hail, damaging winds and heavy downpours. Although downtown Boston has never been struck by a violent tornado, the city itself has experienced many tornado warnings. Damaging storms are more common to areas north, west, and northwest of the city. Boston has a relatively sunny climate for a coastal city at its latitude, averaging over 2,600 hours of sunshine per annum.

Table A.28: Examples of QA examples with answers in *front* group where answers are highlighted.

Answer Position	Question	Context
Back	How many murders did Detroit have in 2015?	Detroit has struggled with high crime for decades. Detroit held the title of murder capital between 1985-1987 with a murder rate around 58 per 100,000. Crime has since decreased and, in 2014, the murder rate was 43.4 per 100,000, lower than in St. Louis, Missouri. Although the murder rate increased by 6% during the first half of 2015, it was surpassed by St Louis and Baltimore which saw much greater spikes in violence. At year-end 2015, Detroit had ***295*** criminal homicides, down slightly from 299 in 2014.
Back	Who was leading the Conservatives at this time?	Despite being a persistent critic of some of the government's policies, the paper supported Labour in both subsequent elections the party won. For the 2005 general election, The Sun backed Blair and Labour for a third consecutive election win and vowed to give him öne last chanceẗo fulfil his promises, despite berating him for several weaknesses including a failure to control immigration. However, it did speak of its hope that the Conservatives (led by ***Michael Howard***) would one day be fit for a return to government. This election (Blair had declared it would be his last as prime minister) resulted in Labour's third successive win but with a much reduced majority.
Back	Who lost the 2015 Nigerian presidential election?	Nigeria is a Federal Republic modelled after the United States, with executive power exercised by the president. It is influenced by the Westminster System model[citation needed] in the composition and management of the upper and lower houses of the bicameral legislature. The president presides as both Head of State and head of the national executive; the leader is elected by popular vote to a maximum of two 4-year terms. In the March 28, 2015 presidential election, General Muhammadu Buhari emerged victorious to become the Federal President of Nigeria, defeating then incumbent ***Goodluck Jonathan***.

Table A.29: Examples of QA examples with answers in *back* group where answers are highlighted.